```
                                                 .SBTTL  SETFIL - CREATE FILTER BLOCK

                                         ;+
                                         ;
                                         ; SETFIL - CREATE FILTER BLOCK
                                         ;
                                         ; INPUTS:
                                         ;
                                         ; OUTPUTS:
                                         ;       CARRY CLEAR:
                                         ;          FILTER BLOCK IS CREATED
                                         ;       CARRY SET:
                                         ;          ERROR MESSAGE DISPLAYED (UNABLE TO CREATE FILTER BLOCK)
                                         ;
                                         ;-

015110  012701  000016                   SETFIL: MOV     #F.LEN,R1               ; GET LENGTH OF BLOCK TO ALLOCATE
015114  032767  000070  000000G                  BIT     #OP$NOD!OP$LNE!OP$SIN,$OPTON ; ANY QUALIFIERS SPECIFIED?
015122  001402                                   BEQ     10$                     ; BR IF NO
015124  012701  000040                           MOV     #F.CEV+10,R1            ; GET LENGTH OF BLOCK TO ALLOCATE
015130  010146                           10$:    MOV     R1,-(SP)                ; SAVE FILTER BLOCK LENGTH
015132                                           CALL    $ALPOL                  ; TRY TO ALLOCATE A FILTER BLOCK
015136  103473                                   BCS     101$                    ; BR IF UNABLE TO ALLOCATE ONE
015140  006000                                   ROR     R0                      ; CHECK IF ALLOCATED FROM POOL..
015142  103071                                   BCC     101$                    ; IF NOT - REPORT ERROR

015144  006100                                   ROL     R0                      ; ADJUST BLOCK ADDRESS
015146  010067  000164'                          MOV     R0,CURUNM               ; SAVE UNMAPPED ADDRESS
015152  012701  000000G                          MOV     #$BFR,R1                ; SET UP FILTER BLOCK
015156  005021                                   CLR     (R1)+                   ; INITIALIZE LINK WORD
015160  016711  000002G                          MOV     RQB+LO.CLS,(R1)         ; STORE EVENT CLASS
015164  051621                                   BIS     (SP),(R1)+              ; STORE LENGTH OF FILTER BLOCK
015166  032767  000000G 000000G                  BIT     #CM$SET,$CMAND          ; SET COMMAND?
015174  001011                                   BNE     20$                     ; BR IF YES
015176                                           CALL    INIEVT                  ; INITIALIZE THE SET EVENT MASKS
015202  016721  000120'                          MOV     EVQUAL,(R1)+            ; STORE QUALIFIER FLAGS
015206                                           CALL    SETQLF                  ; SET QUALIFIERS
015212                                           CALL    STREVT                  ; STORE THE CLEAR EVENT MASKS
015216  000414                                   BR      30$                     ; CONTINUE

015220                                   20$:    CALL    STREVT                  ; STORE THE EVENT MASKS
015224  016721  000120'                          MOV     EVQUAL,(R1)+            ; STORE FLAGS WORD
015230  032767  000071  000000G                  BIT     #OP$NOD!OP$LNE!OP$SIN!OP$HOS,$OPTON ; ANY QUALIFIERS?
015236  001404                                   BEQ     30$                     ; BR IF NO
015240                                           CALL    SETQLF                  ; STORE QUALIFIER FIELDS
015244                                           CALL    INIEVT                  ; INITIALIZE CLEAR EVENA MASK FIELD
                                         ;
                                         ; LINK NEW FILTER BLOCK INTO LIST - SORT BY SINK NODE, THEN CLASS
                                         ;
015250                                   30$:    CEACC$  R0,R1                   ; CONVERT TO MAPPED ADDRESS
015260                                   40$:    PUTAD   R1,(SP)+                ; WRITE OUT FILTER BLOCK
015274  016701  000014G                          MOV     $BFR+F.FLG,R1           ; GET FLAGS WORD
015300  016702  000002G                          MOV     $BFR+F.CLS,R2           ; GET EVENT CLASS NUMBER
015304  016703  000026G                          MOV     $BFR+F.REM,R3           ; GET SINK NODE
015310                                           CALL    FINFLT                  ; FIND PLACE FOR FILTER BLOCK
015314  012702  000000G                          MOV     #$FILHD,R2              ; POINT TO FILTER LISTHEAD
015320                                           CALL    QINS                    ; AND INSERT ENTRY INTO LIST
```

```
  3787                                                .SBTTL  SAVFIL - SAVE FILTER MASKS
  3788
  3789                                        ;+
  3790                                        ;
  3791                                        ; SAVFIL - COPY FILTER MASKS TO SAVE AREA
  3792                                        ;
  3793                                        ; INPUTS:
  3794                                        ;       R3 - POINTER TO FILTER MASKS TO SAVE
  3795                                        ;       R5 - POINTER TO SAVE AREA
  3796                                        ;
  3797                                        ; OUTPUTS:
  3798                                        ;       THE EVENT FILTER MASKS ARE SAVED
  3799                                        ;       R3,R5 DESTROYED
  3800                                        ;
  3801                                        ;-
  3802
  3803 017376  012325                         SAVFIL: MOV     (R3)+,(R5)+             ; SAVE THE FILTER MASKS
  3804 017400  012325                                 MOV     (R3)+,(R5)+             ; ...
  3805 017402  012325                                 MOV     (R3)+,(R5)+             ; ...
  3806 017404  012325                                 MOV     (R3)+,(R5)+             ; ...
  3807 017406                                         RETURN
```

```
                                        .SBTTL  $VLTM - SET/CLEAR LISTEN TIMER

                                ;+
                                ;
                                ; $VLTM - SET LISTEN TIMER
                                ;
                                ; INPUTS:
                                ;       LO.LTM+RQB - SPECIFIED LISTEN TIMER
                                ;
                                ; OUTPUTS:
                                ;       LISTEN TIMER STORED IN COUNTER BLOCK
                                ;
                                ;-
020736                          $VLTM:: CALL    FNDCTR                  ; FIND COUNTER BLOCK
020742  103406                          BCS     101$                    ; BR IF CAN'T FIND IT
020744  016767  000034G 000032G         MOV     LO.LTM+RQB,T$T6+$BFR    ; STORE LISTEN TIMER
020752                                  PUTAD                           ; REWRITE COUNTER BLOCK
020756                                  RETURN
020760                          101$:   ERRPT$  ERR44,$EROUT            ; PARAMATER NOT APPLICABLE
```

```
                                                         B  4
VENA - VNP ENABLE/DISABLE LINES MACRO V05.03b Wednesday 11-Sep-85 12:06  Page 85-6
Symbol table

U2.DJ1= 040000        VF.FIX= ****** GX     $CAT5 = ****** GX     $OPTON= ****** GX     $VHST   020662RG
U2.DZ1= 000100        VF.ON = ****** GX     $CEACC= ****** GX     $PDVNM= ****** GX     $VHTM   020770RG
U2.ESC= 001000        VF.PEM= ****** GX     $CMAND= ****** GX     $PDVTA= ****** GX     $VINA   004144RG
U2.HFF= 010000        VF.SED= ****** GX     $CNOD   003466RG      $PUTAD= ****** GX     $VINC   004054RG
U2.HLD= 000040        VF.SKA= ****** GX     $COTB = ****** GX     $PUTRC= ****** GX     $VLIN   004234RG
U2.LOG= 000400        VF.SS = ****** GX     $DEA16= ****** GX     $QUAL = ****** GX     $VLOG   012002RG
U2.LWC= 000001        V$FLG   000000        $DECPT= ****** GX     $RADDR= ****** GX     $VLTM   020736RG
U2.L3S= 000004        V$LEN   000022        $DEPOL= ****** GX     $RBLCK= ****** GX     $VOBJ   010306RG
U2.L8S= 010000        V$RCV   000002        $DEVHD= ****** GX     $RDBLH= ****** GX     $VOUT   004110RG
U2.NEC= 004000        V$XMT   000012        $EROUT= ****** GX     $RPASW  006130RG      $VOWN   001510RG
U2.PRV= 000010        V$$CTR= 001000        $FILHD= ****** GX     $RSIZE= ****** GX     $VREM   004654RG
U2.RMT= 020000        WC.CNT= 000400        $FNADD= ****** GX     $SAVAL= ****** GX     $VRET   004200RG
U2.R04= 100000        WC.EVE= 004000        $FNALI= ****** GX     $SAVRG= ****** GX     $VSEG   004020RG
U2.SCS= 000004        WC.TRI= 002000        $FNNAM= ****** GX     $SLN  = ****** GX     $VSER   000242RG
U2.SLV= 000200        WC.UNT= 001000        $FNOBJ= ****** GX     $SLTA = ****** GX     $VTAD   001230RG
U2.VT5= 000002        X$$DBT= 000000        $FPDV   021716RG      $SLTMA= ****** GX     $XLINK= ****** GX
U2.7CH= 010000        X25NAM  000004R   002 $GETAD= ****** GX     $SMOD   007524RG      $$    = 177777
U3.OPA= 100000        X5DSMT  020536R       $GETRV= ****** GX     $SNOD   003460RG      $$LCNT= 000072 G
U3.PAR= 040000        ZF.LLC= ****** GX     $HMBUF= ****** GX     $TPASW  006260RG      $$LLIN= 000104 G
U3.UPC= 020000        Z.DAT = ****** GX     $LGCON= ****** GX     $TRIB = ****** GX     $$$   = 003632R
U4.CR = 000100        Z.FLG = ****** GX     $LGFNB= ****** GX     $VACP   006524RG      $$$$  = 003656R
VALI    011734R       Z.LLN = ****** GX     $LGMON= ****** GX     $VACT   001350RG      .ABTIM= ****** GX
VF$RCV= 100000        Z.MAP = ****** GX     $LGSTT= ****** GX     $VALI   011012RG      .CLKHD= ****** GX
VF$XMT= 040000        Z.NAM = ****** GX     $LLCTA= ****** GX     $VCOST  002006RG      .MGMGE= ****** GX
VF.ACP= ****** GX     Z.PCB = ****** GX     $LSTAD= ****** GX     $VDEA   001430RG      .TLGTH= ****** GX
VF.ADD= ****** GX     $ALL15= ****** GX     $LSTRM= ****** GX     $VENA   000250RG      .TSKHD= ****** GX
VF.DLX= ****** GX     $ALPOL= ****** GX     $MUL  = ****** GX     $VERIF  006410RG      ..COO = ****** GX
VF.EAD= ****** GX     $BFR  = ****** GX     $OBJHD= ****** GX     $VFLAG= ****** GX     ..OFF.= 177732


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        021770    001   (RW,I,LCL,REL,CON)
DATA    003430    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 271
Work  file writes: 225
Size of work file: 41075 Words  ( 161 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:05:04.14
DB2:VENA.T34,[132,134]VENA/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VENA
```

```
                                                         C  4
VENA          CREATED BY  MACRO  ON 11-SEP-85 AT 12:08        PAGE 1
```

```
                                                                  B  5
VENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 13

SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE             REFERENCES

RE$DRP  = 000016          #5-79
RE$LDT  = 000013          #5-79
RE$LSN  = 000012          #5-79
RE$NML  = 000001          #5-79
RE$OPE  = 000004          #5-79
RE$OPR  = 000000          #5-79
RE$RCV  = 000001          #5-79
RE$SED  = 000011          #5-79
RE$SKW  = 000006          #5-79
RE$STA  = 000002          #5-79
RE$SUM  = 000003          #5-79
RE$SYN  = 000000          #5-79
RE$TME  = 000021          #5-79
RE$TMO  = 000000          #5-79
RE$TMR  = 000020          #5-79
RE$UPT  = 000002          #5-79
RE$URE  = 000003          #5-79
RE$VER  = 000005          #5-79
RE$VRQ  = 000015          #5-79
RF.LOO  = 100000          #5-74      27-1433    27-1456
RQB     = ******   GX      13-597    13-602     13-607     14-640     15-672     16-696     17-783     17-797     26-1269
                           26-1271   26-1273    26-1275    26-1291    26-1308    26-1325    26-1342    26-1359    27-1380
                           27-1432   27-1441    27-1450    27-1455    28-1492    28-1498    28-1506    28-1514    28-1526
                           28-1530   28-1535    28-1539    28-1559    28-1564    36-1841    37-1879    38-1916    41-2197
                           41-2233   41-2235    41-2241   *41-2246    41-2248    41-2251    41-2252    41-2263    42-2301
                           42-2367   42-2370    42-2378    42-2380    42-2382    42-2395    42-2398    42-2399    42-2402
                           42-2406   42-2410    45-2560    45-2603    46-2641    47-2685    50-2821    50-2829    51-2938
                           53-3024   54-3060   *56-3140    56-3141    56-3146    57-3194    57-3213    59-3321    62-3491
                           62-3498   62-3511    62-3552    62-3555    62-3560    62-3567    62-3572    62-3601    69-3850
                           69-3851   69-3852    69-3853    71-3900    78-4183    79-4207    80-4227
RT$INI  = 000002          #5-79
RT$OFF  = 000001          #5-79
RT$ON   = 000000          #5-79
RUN       021232 R         39-2017   40-2159   #82-4305
RX.NXM  = 000020          #5-82
RX.OVR  = 000010          #5-82
R$$EIS  = ******           46-2642   56-3139
R$$MPL  = ******           39-1958   39-1970    40-2101    40-2117
R.ADD     000010          #5-74      27-1389    27-1427   *27-1432    27-1435   *27-1455   *28-1539   *28-1564    28-1576
                           30-1650   53-3033    54-3067
R.CSR     000056          #5-82
R.FLAG    000012          #5-74      27-1391    27-1428   *27-1433   *27-1456    28-1495    28-1509   *28-1565
R.FLG     000064          #5-82
R.LEN     000014          #5-74      27-1415    27-1444    28-1553    28-1570    30-1649
R.LNK     000000          #5-74
R.LST     000104          #5-82      21-977     21-979
R.MAP     000110          #5-82
R.NAM     000002          #5-74      27-1449    27-1454    28-1513    28-1558    28-1563    42-2371
R.QUE     000062          #5-82
R.SLN     000070          #5-82
R.SRV     000060          #5-82
R.STA     000066          #5-82
```

```
                                                                  C  5
VENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 14
```

```
        .TITLE  VEXP - VNP EXPAND TEMPLATE PROCESSING
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - LINE TABLE EXPANSION FRONT END
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
000406  156724  000040G                 BISB    D$MLL+$HMBUF,(R4)+; COPY LOGICAL LINK COUNT
000412  105024                          CLRB    (R4)+              ; INIT HIGH BYTE
000414                                  SWTCHI                     ; RESET TO DEFAULT BUFFER $BFR
000422                                  RETURN


                                ;
                                ; LISTHEAD
                                ;
000424  010400                  LSTHD:  MOV     R4,R0              ; SET REAL ADDRESS
000426  066700  000000G                 ADD     $LINK,R0           ;   ...
000432  005024                          CLR     (R4)+              ; STORE A ZERO WORD
000434  010024                          MOV     R0,(R4)+           ; STORE ADDRESS OF PREVIOUS WORD
000436                                  RETURN


                                ;
                                ; LINE TABLE ADDRESS
                                ;
000440  016724  000000G         LTAB:   MOV     $SADDR,(R4)+       ; STORE LINE TABLE STARTING ADDRESS
000444                                  RETURN


                                ;
                                ; SECONDARY CSR
                                ;
000446  005767  000450'         SECSR:  TST     FLAGS              ; IS IT MUX EXPANSION?
000452  100015                          BPL     101$               ; IF PL, NO
000454  016700  000446'                 MOV     INDEX,R0           ; GET SECONDARY FILE COUNTER
000460  032767  000000G 000000G         BIT     #FL.KMX,$FLAGS     ; IS THIS A COMIOP MUX LINE?
000466  001403                          BEQ     10$                ; IF EQ, NO
000470  006200                          ASR     R0                 ; DIVIDE COUNTER BY 8
000472  006200                          ASR     R0                 ; (NUMBER OF LINES PER CONTROLLER)
000474  006200                          ASR     R0                 ;   ...
000476  006300                  10$:    ASL     R0                 ; CONVERT TO A WORD INDEX
000500  016024  000000G         20$:    MOV     $$SCSR(R0),(R4)+   ; COPY SECONDARY CSR ADDRESS
000504                                  RETURN
000506  012724  177777          101$:   MOV     #-1,(R4)+          ; SECSR NOT IN MXTAB FILE
000512                                  RETURN


                                ;
                                ; SYSTEM LINE NUMBER (BYTE)
                                ;
000514  005000                  SLNB:   CLR     R0                 ; ASSUME NORMAL EXPANSION
000516  005767  000450'                 TST     FLAGS              ; IS IT MUX EXPANSION?
000522  100003                          BPL     10$                ; IF PL, NO
000524  016700  000446'                 MOV     INDEX,R0           ; GET MUX INDEX
000530  122020                          CMPB    (R0)+,(R0)+        ; SKIP DLC LINE NUMBER
000532  116024  000000G         10$:    MOVB    $SLN(R0),(R4)+     ; STORE SYSTEM LINE NUMBER
000536                                  RETURN


                                ;
                                ; SYSTEM LINE NUMBER (WORD)
                                ;
000540  005000                  SLNW:   CLR     R0                 ; ASSUME NORMAL EXPANSION
000542  005767  000450'                 TST     FLAGS              ; IS IT MUX EXPANSION?
000546  100003                          BPL     10$                ; IF PL, NO
000550  016700  000446'                 MOV     INDEX,R0           ; GET MUX INDEX
000554  122020                          CMPB    (R0)+,(R0)+        ; SKIP DLC LINE NUMBER
000556  005001                  10$:    CLR     R1                 ; GET LINE NUMBER AS A WORD
```

```
                                        .SBTTL  X2CH
                                ;
                                ; MAXIMUM WINDOW SIZE
                                ;
002342  116724  000000G         X2CH01: MOVB    $X2MWS,(R4)+            ; STORE MAXIMUM WINDOW SIZE
002346                                  RETURN
                                ;
                                ; MAXIMUM BLOCK SIZE
                                ;
002350  016724  000000G         X2CH02: MOV     $X2MBS,(R4)+            ; STORE MAXIMUM BLOCK SIZE
002354                                  RETURN
                                ;
                                ; MAXIMUM RETRANSMIT COUNT
                                ;
002356  116724  000000G         X2CH03: MOVB    $X2MRT,(R4)+            ; STORE MAXIMUM RETRANSMIT COUNT
002362                                  RETURN
                                ;
                                ; RETRANSMIT TIMER
                                ;
002364  016724  000000G         X2CH04: MOV     $X2RET,(R4)+            ; STORE RETRANSMIT TIMER
002370                                  RETURN
                                ;
                                ; HOLDBACK TIMER
                                ;
002372  016724  000000G         X2CH05: MOV     $X2HLD,(R4)+            ; STORE HOLDBACK TIMER
002376                                  RETURN

        000001                          .END
```

```
                                        .SBTTL  $VEX3 - TEMPLATE EXPANSION CLEANUP
                                ;+
                                ; $VEX3 - CLEANUP BINARY INFORMATION AFTER TEMPLATE ERROR
                                ;
                                ; THIS ROUTINE IS CALLED ONLY IF A TEMPLATE ERROR OCCURS. IT DEALLOCATES
                                ; ALLOCATIONS AND LIBRARY REFERENCES MADE EARLIER IN THE TEMPLATE.
                                ;
                                ; INPUTS:
                                ;       $BIN=START OF BINARY INFORMATION
                                ;       $PTR=END OF BINARY INFORMATION
                                ;
                                ; OUTPUTS:
                                ;       ALL REGISTERS=DESTROYED
                                ;-
000000                          $VEX3::
000000  016705  000000G                 MOV     $BIN,R5         ; START OF BINARY DATA
000004  016704  000000G                 MOV     $ADDR,R4        ; START OF REAL DATA
000010  026705  000000G         10$:    CMP     $PTR,R5         ; IS IT END OF BINARY DATA?
000014  001407                          BEQ     20$             ; IF EQ, YES
000016  005003                          CLR     R3              ; GET FUNCTION CODE
000020  152503                          BISB    (R5)+,R3        ; ...
000022  001404                          BEQ     20$             ; IF ZERO, END OF FILE
000024  106303                          ASLB    R3              ; CONVERT TO A WORD INDEX
000026                                  CALL    @TABLE-2(R3)    ; EXPAND ONE FIELD
000032  000766                          BR      10$             ; KEEP LOOPING UNTIL END
000034                          20$:    RETURN
```

```
                                                .SBTTL  $VEX5 - TEMPLATE EXPANSION
                                        ;+
                                        ; $VEX5 - EXPAND BINARY INFORMATION INTO REAL INFORMATION
                                        ;
                                        ; THIS ROUTINE IS CALLED AFTER THE ASCII TEMPLATE FILE HAS BEEN READ
                                        ; AND PROCESSED INTO THE BINARY FILE. THE BINARY INFORMATION IS EXPANDED
                                        ; AND STORED AT THE LOCATION CONTAINED IN "$ADDR".
                                        ;
                                        ; INPUTS:
                                        ;       $BIN=START OF BINARY INFORMATION
                                        ;       $PTR=END OF BINARY INFORMATION
                                        ;       $ADDR=ADDRESS OF WHERE REAL DATA GOES
                                        ;
                                        ; OUTPUTS:
                                        ;       ALL REGISTERS=DESTROYED
                                        ;-
000000                                  $VEX5::
000000  005067  000152'                         CLR     INDEX           ; ZERO INDEX VALUE
000004  005067  000154'                         CLR     FLAGS           ; AND SPECIAL FLAGS
000010  016705  000000G                         MOV     $BIN,R5         ; START OF BINARY DATA
000014  016703  000000G                         MOV     $ADDR,R3        ; GET LINE TABLE ADDRESS
000020  016704  000000G                         MOV     $SLTA,R4        ; GET THE SLT ADDRESS
000024  116404  000002                          MOVB    L.DDM(R4),R4    ; GET DDM PDV INDEX
000030                                          CALL    FLT16           ; SEARCH THRU THE UNLOAD BLOCKS
000034  016702  000160'                         MOV     UBA,R2          ; SUCCESS ?
000040  001417                                  BEQ     101$            ; NO ..
000042  016704  000000G                         MOV     $SLTA,R4        ; RESTORE THE SLT ADDRESS
000046  016703  000162'                         MOV     CTRADD,R3       ; GET CONTROLLER TABLE STARTING ADDRESS
000052  026705  000000G                 10$:    CMP     $PTR,R5         ; IS IT END OF BINARY DATA?
000056  001407                                  BEQ     20$             ; IF EQ, YES
000060  005002                                  CLR     R2              ; GET FUNCTION CODE
000062  152502                                  BISB    (R5)+,R2        ;      ...
000064  001404                                  BEQ     20$             ; IF ZERO, END OF FILE
000066  106302                                  ASLB    R2              ; CONVERT TO A WORD INDEX
000070                                          CALL    @TABLE-2(R2)    ; EXPAND ONE FIELD
000074  000766                                  BR      10$             ; KEEP LOOPING UNTIL END
000076                                  20$:    RETURN

                                        ;
                                        ; ERROR CONDITION
                                        ;
000100                                  101$:   CRASH                   ; CAN'T FIND LINE TABLE IN UNLOAD BLOCKS
```

```
VEX5 - VNP UPDATE LINE TABLES   MACRO V05.03b Saturday 29-Jun-85 02:28  Page 12-1
Symbol table

ADDRB    000102R        INT      000506R        LTAB     000600R        PRI      000606R        V$$CTR= 001000
ADDRW    000106R        I$$RAR= 000000          L$$ASG= 000000          PRTJV    000620R        WORD     001034R
APR      001450R        I$$RDN= 000000          L$$DRV= 000000          P$$P45= 000000          X$$DBT= 000000
A$$CHK= 000000          K$$CNT= 177546          L$$P11= 000001          P$$WRD= 000000          X2CH01   001750R
A$$CPS= 000000          K$$CSR= 177546          L$$11R= 000000          Q$$OPT= 000010          X2CH02   001754R
A$$PRI= 000000          K$$LDC= 000000          L.COST   000015         R$$DER= 000000          X2CH03   001760R
A$$TRP= 000000          K$$TPS= 000074          L.CTL    000012         R$$K11= 000001          X2CH04   001764R
BLKB     000112R        LD$LP  = 000000         L.CVA    177776         R$$SND= 000000          X2CH05   001770R
BLKW     000130R        LFILE    001160R        L.DDM    000002         R$$11M= 000000          X3CH01   001674R
BLOCK    000164R    002 LF.ACT= 100000          L.DDS    000004         SCOM     000626R        X3CH02   001700R
BYTE     000150R        LF.BRO= 000400          L.DLC    000003         SECSR    000640R        X3CH03   001704R
CORE     000156R        LF.BWT= 000007          L.DLM    000006         SF.ACT= 000200          X3CH04   001710R
CSR      000210R        LF.ENA= 002000          L.DLS    000010         SF.ENA= 000100          X3CH05   001714R
CTIM     000220R        LF.LPB= 001000          L.FLG    000000         SF.LPB= 000004          X3CH06   001720R
CTRADD   000162R    002 LF.MDC= 000100          L.KRBA   000016         SF.MFL= 000040          X3CH07   001724R
C$$CKP= 000000          LF.MFL= 004000          L.LEN  = 000022         SF.PAC= 000020          X3CH08   001730R
C$$ORE= 000400          LF.MTP= 000020          L.MPF    000022         SF.REA= 000010          X3CH09   001734R
C$$RSH= 177564          LF.PAC= 000200          L.NMST   000020         SF.SER= 000001          X3CH10   001740R
DPRB     001450R        LF.RDY= 040000          L.NSTA   000014         SF.SVC= 000002          X3CH11   001744R
DPRW     001454R        LF.REA= 010000          L.OWNR   000021         SF.UNL= 000[illegible]  $ADDR  = ****** GX
DVCHA    000224R        LF.SER= 000040          L.UNT    000013         SLNB     0007[illegible]6R  $ADRX  = ****** GX
D$$BUG= 177514          LF.TIM= 000010          MPTAB    001156R        SLNW     001002R        $ADRZ  = ****** GX
D$$ISK= 000000          LF.UNL= 020000          MXPTB    001242R        STNB     001010R        $BFR   = ****** GX
D$$L11= 000001          LF.X2P= 000000          MXTAB    001044R        STNW     001014R        $BIAS  = ****** GX
D$$YNC= 000000          LIBR     000532R        M$$CRB= 000124          SYSFDB= ****** GX       $BIN   = ****** GX
D$$YNM= 000000          LINKS    000564R        M$$CRX= 000000          S$$WRG= 000000          $FLAGS= ****** GX
EOF      000504R        LN.CLO= 000000          M$$FCS= 000000          S$$YSZ= 007600          $GETAD= ****** GX
E$$XPR= 000000          LN.DUM= 000005          M$$MGE= 000000          S.COST   000001         $LINK  = ****** GX
FILE     001050R        LN.LOA= 000004          M$$NET= 000000          S.FLG    000000         $NTLHB= ****** GX
FLAGS    000154R    002 LN.LOO= 000003          M$$OVR= 000000          S.LEN    000004         $PTR   = ****** GX
FLT      001460R        LN.OAU= 000003          NOD      000614R        S.NMST   000002         $PUTAD= ****** GX
FLT16    001460R        LN.OFF= 000001          N$$ACC= 000001          S.OWNR   000003         $RSIZE= ****** GX
FL.KMX= ****** GX       LN.ON  = 000000         N$$BUF= 000001          TABLE    000000R    002 $SLTA  = ****** GX
F$$LVL= 000001          LN.OOP= 000004          N$$LDV= 000001          TIME     001444R        $VEX5    000000RG
F.RSIZ= ****** GX       LN.OPE= 000001          N$$MCP= 000001          T$$KMG= 000000          $$DCHA= ****** GX
F.URBD= ****** GX       LN.REF= 000002          N$$MLL= 000001          T$$MIN= 000000          $$PCHA= ****** GX
G$$TPP= 000000          LN.SER= 000002          N$$MOV= 000010          UBA      000160R    002 $$SCSR= ****** GX
G$$TSS= 000000          LN.STA= 000017          N$$NCT= 000001          UBIAS    000156R    002 .CXUNL= ****** GX
G$$TTK= 000000          LN.SUB= 000360          N$$PEM= 000001          UNB      001022R        .EOF   = ****** GX
G$$WRD= 000000          LN.TRI= 000006          FECHA    000370R        UNW      001026R        .MGMGE= ****** GX
INDEX    000152R    002 LSTHD    000572R

. ABS.   177776     000    (RW,I,GBL,ABS,OVR)
         001774     001    (RW,I,LCL,REL,CON)
DATA     000264     002    (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11239 Words  ( 44 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:19.19
SY:VEX5.V2,[132,134]VEX5/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VEX5
```

```
                              .SBTTL  TPARS STATE TABLES
                      ;
                      ; TPARS STATE TABLES
                      ;
000052                        ISTAT$  DTEST,DTEKW
                      ;
                      ; DTE$DF
                      ;
000052                        STATE$  DTEDF
000052                          TRAN$   %DTE$DF%,,,1,SYNERR

000052                        STATE$
000052                          TRAN$   $DIGIT,,DTEAST          ; START OF DTE ADDRESS

000052                        STATE$  DTEAD
000052                          TRAN$   $DIGIT,DTEAD,DTEDIG     ; DTE ADDRESS
000052                          TRAN$   <',>,,DTEAEN            ; END OF DTE ADDRESS

000052                        STATE$
000052                          TRAN$   !LINID                  ; LINE-ID

000052                        STATE$
000052                          TRAN$   <',>

000052                        STATE$
000052                          TRAN$   $NUMBR,,HSHTBL          ; HASH TABLE SIZE

000052                        STATE$
000052                          TRAN$   <',>

000052                        STATE$
000052                          TRAN$   $NUMBR,,CNTTIM          ; COUNTER TIMER VALUE

000052                        STATE$
000052                          TRAN$   !END,$EXIT,DTEEND
000052                          TRAN$   <',>

000052                        STATE$
000052                          TRAN$   "ON",DTENW,,DE.ON,DTEFLG        ; STATE
000052                          TRAN$   "OFF",DTENW,,DE.OFF,DTEFLG

000052                        STATE$  DTENW
000052                          TRAN$   !END,$EXIT,DTEEND
000052                          TRAN$   <',>

000052                        STATE$
000052                          TRAN$   $STRNG,,DTENET          ; NETWORK NAME PARAMETER

000052                        STATE$  END
000052                          TRAN$   $EOS,$EXIT
000052                          TRAN$   <':>,$EXIT
```

```
VFGDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:28      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE             REFERENCES

BIASX      000576 R        #17-417
CERR     = ******  GX       8-128      8-129      8-130      8-131      8-132
CFGBF    = ******  GX       9-156
CFGSZ    = ******  GX       9-155
CFLIN    = ******  GX       8-128      8-129      8-130      8-131      8-132
CHKDTE     000426 R         15-353   #16-377
CNTTIM     000332 R        #13-321
COUNT      000035 R        #7-118    *12-254    *19-478     20-498
CVTBUF     000036 R        #7-119    *19-470     19-474
DEVCTL     000644 R        #18-438
DEVNAM     000616 R        #18-429
DEVUNT     000662 R        #18-445
DTEADD     000000 R        #7-113     12-250     12-255     12-276
DTEAEN     000140 R        #12-274
DTEALN   = 000017          #7-112      7-113     12-249     12-275     16-390
DTEAST     000052 R        #12-248
DTEDES   = ******  GX      *16-386
DTEDF      000000 R         9-152
DTEDIG     000114 R        #12-262
DTEEND     000410 R         14-342   #15-353
DTEFLG   = ******  GX      *12-248
DTEKW      000000 RG        9-154    #10-173
DTEND    = 000016 R        #7-114     12-262
DTENET     000350 R        #14-334
DTEPCK     000017 R        #7-115     12-277     16-393
DTEST      000000 RG       #10-173
DTNMMX   = 000006          #7-116      7-117     12-281     12-288     12-291     14-335
DTNTNM     000027 R        #7-117     12-280     12-290     14-338
FMT8B    = ******  GX       8-128      8-129      8-130      8-131      8-132
FM.8B    = 000000          #8-128    #8-129     #8-130     #8-131     #8-132
F.RSIZ   = ******  GX      *12-288   *16-380    *16-389
HSHMN    = 000040          #6-85      13-304
HSHMX    = 001000          #6-86      13-308
HSHSZ    = ******  GX      *13-311
HSHTBL     000262 R        #13-301
H$LDTE     000002           16-379
H$NETW     000024           12-287
I$$AS    = ******           20-508
LF.ACT   = 100000          #5-67
LF.BRO   = 000400          #5-67
LF.BWT   = 000007          #5-67
LF.ENA   = 002000          #5-67
LF.LPB   = 001000          #5-67
LF.MDC   = 000100          #5-67
LF.MFL   = 004000          #5-67
LF.MTP   = 000020          #5-67
LF.PAC   = 000200          #5-67
LF.RDY   = 040000          #5-67
LF.REA   = 010000          #5-67
LF.SER   = 000040          #5-67
LF.TIM   = 000010          #5-67
LF.UNL   = 020000          #5-67
```

```
    208                                         .SBTTL  TPARS STATE TABLES
    209                                 ;
    210                                 ; TPARS STATE TABLES
    211                                 ;
    212 000056                                  ISTAT$  PVCST,PCVKW
    213
    214                                 ;
    215                                 ; PVC$DF STATE TABLES
    216                                 ;
    217 000056                                  STATE$  PVCDF
    218 000056                                   TRAN$   %PVC$DF%
    219
    220 000056                                  STATE$
    221 000056                                   TRAN$   $RAD50,,PVNAM           ; CHANNEL IDENTIFICATION
    222
    223 000056                                  STATE$
    224 000056                                   TRAN$   !CHKDLM,CHNLNO
    225
    226 000056                                  STATE$  CHNLNO
    227 000056                                   TRAN$   $NUMBR,,PVLCN,1,SYNERR ; LOGICAL CHANNEL NUMBER
    228
    229 000056                                  STATE$
    230 000056                                   TRAN$   <',>
    231
    232 000056                                  STATE$
    233 000056                                   TRAN$   $NUMBR,,PVCT            ; COUNTER TIMER
    234
    235 000056                                  STATE$
    236 000056                                   TRAN$   <',>
    237
    238 000056                                  STATE$
    239 000056                                   TRAN$   ''OFF'',PSZ              ; STATE
    240 000056                                   TRAN$   ''ON'',PSZ
    241
    242 000056                                  STATE$  PSZ
    243 000056                                   TRAN$   !END,$EXIT,PVEND
    244 000056                                   TRAN$   <',>
    245
    246 000056                                  STATE$
    247 000056                                   TRAN$   $NUMBR,WNDSZ,PVPSZ      ; PACKET SIZE
    248 000056                                   TRAN$   $LAMDA
    249
    250 000056                                  STATE$  WNDSZ
    251 000056                                   TRAN$   !END,$EXIT,PVEND
    252 000056                                   TRAN$   <',>
    253
    254 000056                                  STATE$
    255 000056                                   TRAN$   $NUMBR,OWNER,PVWSZ      ; WINDOW SIZE
    256 000056                                   TRAN$   $LAMDA
    257
    258 000056                                  STATE$  OWNER
    259 000056                                   TRAN$   !END,$EXIT,PVEND
    260 000056                                   TRAN$   <',>
    261
    262 000056                                  STATE$
    263 000056                                   TRAN$   $RAD50,PFLAGS,PVOWN     ; PROCESS OWNER
    264 000056                                   TRAN$   $LAMDA
```

```
VFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:29  Page 20-2
Symbol table

ALLXCB  000434R        E$NSEG  000010         I$$RDN= 000000         L$CLEN= 000044         L.UNT   000013
A$$CHK= 000000         E$NTIM  000046         K$$CNT= 177546         L$CTEN  000032         MAXID = 000020
A$$CPS= 000000         E$NUSE  000004         K$$CSR= 177546         L$CTIM  000040         M$$CRB= 000124
A$$PRI= 000000         E$STRT  000006         K$$LDC= 000000         L$DTEA  000020         M$$CRX= 000000
A$$TRP= 000000         E$$XPR= 000000         K$$TPS= 000074         L$DTEL  000017         M$$FCS= 000000
BIASX   001466R        FLAG    000032R    002 LD$LP = 000000         L$GLEN  000134         M$$MGE= 000000
CERR  = ****** GX      FMT8  = ****** GX      LF.ACT= 100000         L$LEN   000122         M$$NET= 000000
CFGBF = ****** GX      FMT8B = ****** GX      LF.BRO= 000400         L$LLCH  000016         M$$OVR= 000000
CFGSZ = ****** GX      FM.8  = 000000         LF.BWT= 000007         L$LNK   000000         N$ADJ1  000072
CFLIN = ****** GX      FM.8B = 000000         LF.ENA= 002000         L$MCHN  000036         N$ADJ2  000074
CF$BLK= 000102         FNDHSH  001304R        LF.LPB= 001000         L$NETW  000114         N$CACH  000062
CHKBLK= ****** GX      FNDPDV= ****** GX      LF.MDC= 000100         L$NIRE  000072         N$CRC   000120
CHKDLM  000164R    003 FNDPRT  000664R        LF.MFL= 004000         L$NLRE  000104         N$HC1   000052
CHKWND= ****** GX      F$$LVL= 000001         LF.MTP= 000020         L$NMAC  000076         N$HC2   000056
CHNLMX= ****** GX      F.RSIZ= ****** GX      LF.PAC= 000200         L$NMAS  000074         N$LV1   000002
CHNLNO  000014R    003 G$$TPP= 000000         LF.RDY= 040000         L$NMST  000003         N$LV2   000010
C$DTE   000012         G$$TSS= 000000         LF.REA= 010000         L$NNRE  000106         N$MHC1  000036
C$FLG   000015         G$$TT[illegible]= 000000         LF.SER= 000040         L$NRBY  000046         N$MHC2  000044
C$GNAM  000016         G$$WRD= 000000         LF.TIM= 000010         L$NRCA  000066         N$PLD   000016
C$LCN   000010         HF$DLM= 000002         LF.UNL= 020000         L$NRFS  000100         N$PRI   000076
C$LEN   000016         HF$GWY= 000010         LF.X2P= 000000         L$NRPK  000056         N$ROA1  000022
C$LNK   000000         HF$HOS= 000004         LN$OFF= 000001         L$NRRE  000105         N$ROA2  000030
C$NAM   000002         HF$XDF= 000020         LN$ON = 000000         L$NRST  000107         N$RTMX  000014
C$NML   000002         HSHADD= ****** GX      LN$SHU= 000002         L$NTBY  000052         N$RTM1  000014
C$PORT  000014         HSHSZ = ****** GX      LN.CLO= 000000         L$NTCA  000070         N$RTM2  000015
C$$CKP= 000000         HTIMR   000000R    002 LN.DUM= 000005         L$NTFS  000102         N$RT1   000000
C$$ORE= 000400         H$CUG   000010         LN.LOA= 000004         L$NTPK  000062         N$RT2   000006
C$$RSH= 177564         H$DST   000012         LN.LOO= 000003         L$OMST  000002         N$TCTL  000112
DLMCTL  000057R    002 H$D29   000014         LN.OAU= 000003         L$PCS   000112         N$TLC   000100
DLMIND  000061R    002 H$FLG   000000         LN.OFF= 000001         L$PLS   000110         N$TNC   000114
DLMPDV  000022R    002 H$GLEN  000104         LN.ON = 000000         L$QCNT  000122         N$TRC   000106
DLMSLN  000030R    002 H$GLT   000044         LN.OOP= 000004         L$QUEH  000124         N$TTCB  000110
DLMSLT  000024R    002 H$GNAM  000050         LN.OPE= 000001         L$QUET  000130         N$VER   000066
DLMSTA  000026R    002 H$GNML= 000020         LN.REF= 000002         L$RTRY  000007         N$XLEN  000124
DLMTLC  001662R        H$GPT   000046         LN.SER= 000002         L$SLN   000004         N$$ACC= 000001
DLMUNT  000060R    002 H$HITS  000034         LN.STA= 000017         L$ST    000005         N$$BUF= 00000
DTEDES= ****** GX      H$HLEN  000044         LN.SUB= 000360         L$TCLZ  000044         N$$LDV= 000001
D$$BUG= 177514         H$LBDA  000070         LN.TRI= 000006         L$TIM   000006         N$$MCP= 000001
D$$ISK= 000000         H$LBDN  000072         LP$EIP= 002000         L$$ASG= 000000         N$$MLL= 000001
D$$L11= 000001         H$LDTE  000002         LP$ENB= 004000         L$$DRV= 000000         N$$MOV= 000010
D$$YNC= 000000         H$LEN   000042         LP$PCT= 001400         L$$P11= 000001         N$$NCT= 000001
D$$YNM= 000000         H$LOTS  000032         LP$TMR= 100000         L$$11R= 000000         N$$PEM= 000001
END     000154R    003 H$NETW  000024         LP$UP = 010000         L.COST  000015         OWNER   000110R    003
E$NBR   000014         H$NML = 000006         LP$WTD= 020000         L.CTL   000012         PCVKW   000000RC   004
E$NBS   000020         H$NPT   000022         LP$WTS= 040000         L.CVA   177776         PFLAGS  000132R    003
E$NCR   000034         H$PTB   000020         LS$GDN= 000004         L.DDM   000002         PF$BLK= 020000
E$NCS   000036         H$PVC   000006         LS$OFF= 000000         L.DDS   000004         PF$CLC= 010000
E$NIC   000044         H$RDTE  000004         LS$STR= 000002         L.DLC   000003         PF$DLM= 100000
E$NLEN  000050         H$RNW   000042         LS$UP = 000003         L.DLM   000006         PF$EIP= 000002
E$NLLA  000012         H$SVC   000036         LS$WT = 000001         L.DLS   000010         PF$ENB= 000001
E$NLNK  000000         H$TRB   000016         LTIMR   000002R    002 L.FLG   000000         PF$FAI= 004000
E$NML   000040         H$XAVL  000100         L$ACHN  000012         L.KRBA  000016         PF$FM1= 000100
E$NMR   000024         H$XBIA  000074         L$APVC  000014         L.LEN = 000022         PF$FM2= 000200
E$NMS   000030         H$X29C  000040         L$ASVC  000010         L.MPF   000022         PF$OFF= 000000
E$NNOD  000002         INIXCB  001062R        L$AUC   000042         L.NMST  000020         PF$RM1= 000020
E$NRT   000042         INVCIR  001546R        L$CHLS  000034         L.NSTA  000014         PF$RM2= 000040
E$NRTP  000005         I$$RAR= 000000         L$CHTB  000030         L.OWNR  000021         PF$RVR= 000010
```

```
                               .SBTTL  MACRO DEFINITIONS

                        ;
                        ; LIBRARY MACROS
                        ;
                               .MCALL  ISTAT$,TRAN$,STATE$,EMSG$R,NTLER$,XPDDB$,CTRDF$,PLBDF$
                               .MCALL  PHBDF$,SVCDF$,ASL$,SLTDF$ GETAD,PUTAD,CEACC$,GETRV

000000                         CTRDF$                          ; DEFINE TRANSPORT LINE COUNTER OFFSETS
000000                         XPDDB$                          ; DEFINE XPT DDB OFFSETS
000000                         PHBDF$                          ; DEFINE PSI HOME BLOCK OFFSETS
000000                         SLTDF$                          ; DEFINE SLT OFFSETS
000000                         SVCDF$                          ; DEFINE SVC OFFSETS
000000                         PLBDF$                          ; PLB OFFSETS

                        ;
                        ;
                        ; LOCAL MACRO DEFINTIONS
                        ;

                        ;
                        ; REJECT TPARS TRANSITION
                        ;
                               .MACRO  REJ$
                               ADD     #2,(SP)         ; RETURN TO CALLER+2
                               CLR     SYNERR          ; INDICATE NO SYNTAX ERROR
                               .ENDM   REJ$
                        ;
                        ; SAVE CURRENT MAPPING
                        ;
                               .MACRO  SAVMAP
                               MOV     $RBLCK,-(SP)
                               .ENDM
                        ;
                        ; RESTORE PREVIOUS MAPPING
                        ;
                               .MACRO  RESMAP
                               MOV     (SP)+,$RBLCK
                               .ENDM
                        ;
                        ; SET UP MAPPING
                        ;
                               .MACRO  MAP     BIAS
                               MOV     BIAS,$RBLCK
                               .ENDM
                        ;
                        ; SET UP APR5 BIAS
                        ;
                               .MACRO  BIAS    REG
                               BIC     #160000,REG     ; CLEAR CURRENT MAPPING BITS
                               BIS     #120000,REG     ; SET UP APR5 MAPPING
                               .ENDM
```

```
  3348 015324                           RETURN
  3349                          ;
  3350                          ; ERROR MESSAGES
  3351                          ;
  3352 015326  005726           101$:   TST     (SP)+                   ; CLEAN UP STACK
  3353 015330                           ERRPT$  ERR38,$EROUT            ; UNABLE TO ALLOCATE FILTER BLOCK
```

```
                                        .SBTTL  SAVCOM - SAVE COMPLEMENT OF FILTER MASKS

                                ;+
                                ;
                                ; SAVCOM - SAVE COMPLEMENT OF FILTER MASKS
                                ;
                                ; INPUTS:
                                ;       R3 - ADDRESS OF EVENT MASKS
                                ;       R5 - ADDRESS OF SCRATCH AREA
                                ;
                                ; OUTPUTS:
                                ;       FILTER MASKS COMPLEMENTS ARE SAVED
                                ;       R3,R5 DESTROYED
                                ;
                                ;-

017410  012315                  SAVCOM: MOV     (R3)+,(R5)              ; SAVE FILTER MASKS
017412  005125                          COM     (R5)+                   ; COMPLEMENT THEM
017414  012315                          MOV     (R3)+,(R5)              ; ...
017416  005125                          COM     (R5)+                   ; ...
017420  012315                          MOV     (R3)+,(R5)              ; ...
017422  005125                          COM     (R5)+                   ; ...
017424  012315                          MOV     (R3)+,(R5)              ; ...
017426  005125                          COM     (R5)+                   ; ...
017430                                  RETURN
```

```
                                                .SBTTL  $VHTM - SET/CLEAR HELLO TIMER

                                        ;+
                                        ;
                                        ; $VHTM - SET HELLO TIMER
                                        ;
                                        ; INPUTS:
                                        ;       LO.HTM+RQB - SPECIFIED HELLO TIMER
                                        ;
                                        ; OUTPUTS:
                                        ;       HELLO TIMER STORED IN COUNTER BLOCK
                                        ;
                                        ;-
 020770                                 $VHTM:: CALL    FNDCTR                          ; FIND COUNTER BLOCK
 020774  103406                                 BCS     101$                            ; BR IF CAN'T FIND IT
 020776  016767  000032G 000030G                MOV     LO.HTM+RQB,T$T5+$BFR            ; STORE LISTEN TIMER
 021004                                         PUTAD                                   ; REWRITE COUNTER BLOCK
 021010                                         RETURN
 021012                                 101$:   ERRPT$  ERR43,$EROUT                    ; PARAMETER NOT APPLICABLE
```

```
                                                                       C  4
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL    VALUE          REFERENCES

ACPNAM     000000 R   #6-118     39-1952
AC$DNT   = 000002     #5-79
AC$X25   = 000001     #5-79
AE$CIR   = 000003     #5-79
AE$LIN   = 000001     #5-79
AE$MOD   = 000004     #5-79
ALPLB      020022 R    39-2022  #74-4002
A.ACC      000020     #5-76      42-2322   42-2377
A.LNK      000000     #5-76
A.NAM      000002     #5-76
A.PUD      000010     #5-76
A.REM      000012     #5-76
A.UCB      000010     #5-76
BF.TRC   = 000004     #5-82
CHAN       000070 R   #6-136    *11-431    11-433
CHFIL      014240 R    7-195    #56-3124
CHKEVT     017352 R    62-3530   62-3592   62-3595   62-3628   65-3754  #66-3776
CHKEXE     003730 R    25-1176  #26-1263
CHKXPT     021516 R    11-412    35-1799   36-1828   37-1868   39-1943   74-4003   75-4101  #83-4373
CHNAM      014666 R    7-196    #58-3244
CHSTA      014142 R    7-194    #55-3089
CHTBLE     000262 R   #7-194     44-2520
CKCON      012772 R    47-2703  #49-2766
CKEVE      012144 R    7-181    #45-2557
CKFIL      013072 R    47-2706  #50-2798
CKMDEV     013634 R    49-2771   50-2817  #52-2974
CKMON      012676 R    47-2698  #48-2728
CKNAME     012536 R    7-182    #47-2682
CKNODE     014070 R    7-184    #54-3060
CKSINK     013746 R    7-183    #53-3019
CLASS      000074 R   #6-138    *56-3146   61-3441   63-3686
CLKQB      000050 R   #6-133     82-4314   82-4347
CLRBIT     005724 R    27-1394   27-1430  #33-1737
CLREVT     017334 R    57-3214   62-3493   62-3499   62-3527   62-3557   62-3569   62-3603   62-3625  #65-3750
CL$ASZ   = 010500     #5-79      10-349
CL$DLL   = 000500     #5-79      5-79      5-79      5-79      5-79      5-79      5-79      5-79      5-79
                       5-79      5-79      10-337

CL$ECL   = 000300     #5-79      5-79      10-331
CL$LDN   = 010400     #5-79      5-79      10-346
CL$MAN   = 000000     #5-79      5-79      5-79      5-79      10-325
CL$PAZ   = 034100     #5-79      5-79      5-79      10-361
CL$PLH   = 034000     #5-79      5-79      5-79      5-79      5-79      10-358
CL$PLL   = 000600     #5-79      10-340
CL$PRT   = 034200     #5-79      10-364
CL$ROU   = 010000     #5-79      5-79      5-79      10-343
CL$SES   = 000200     #5-79      5-79      5-79      10-328
CL$SGE   = 035000     #5-79      5-79      5-79      10-367
CL$SSE   = 035100     #5-79      5-79      5-79      5-79      5-79      10-370
CL$TRN   = 000400     #5-79      5-79      5-79      5-79      5-79      5-79      5-79      5-79      5-79
                       5-79      5-79      5-79      5-79      5-79      5-79      5-79      5-79      5-79
                       5-79      10-334

CL$XL2   = 013700     #5-79
```

```
                                                                       D  4
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 2
```

```
SYMBOL CROSS REFERENCE                                         CREF    04.00

SYMBOL   VALUE               REFERENCES

R.STBL     000210            #5-82
SAVCOM     017410 R           57-3204    62-3566     62-3607    62-3613   #68-3825
SAVFIL     017376 R           62-3554    62-3561    #67-3803
SC$OFF   = 000001            #5-79
SC$      = 000000            #5-79
SC$      = 000003            #5-79
SC$SHU   = 000002            #5-79
SER      = 000001            #6-135      11-393      11-448     11-470
SERBIT     001076 R           11-505    #12-539
SERFLG     000126 R          #6-148     *11-393      11-448     11-470
SETBIT     005774 R           27-1436    27-1442    #34-1765
SETEVT     017322 R           62-3512    62-3574    #64-3726
SETFIL     015110 R           57-3216   #59-3307
SETQLF     017466 R           59-3327    59-3335    #71-3890
SF.ACT   = ******    GX       23-1071    24-1111     24-1116
SF.ENA   = ******    GX       11-487     11-500      23-1081    24-1136    24-1140
SF.LPB   = ******    GX       16-733     33-1748     34-1776
SF.SER   = ******    GX       12-545     12-553      12-558
SINKND     000130 R          #6-149     *53-3036     61-3447    63-3701    71-3907
SNAPX      000124 R          #6-145     *11-405     *11-409     11-426
SNKTYP     000230 R          #6-166      56-3128
SNODTB     003660 R           25-1172   #25-1225
STQUAL     017650 R           56-3125   #73-3959
STREVT     017432 R           59-3328    59-3331    #69-3850
SV$DUM   = 000001            #5-79
SV$LOA   = 000000            #5-79
SYSFDB   = ******    GX      *11-442    *20-919     *20-927    *21-977    *21-984    *22-1031   *22-1037   *26-1267   *26-1290
                             *26-1307   *26-1324    *26-1341   *26-1358   *27-1398   *27-1399   *27-1400   *27-1406   *27-1407
                             *27-1408   *28-1570    *28-1572   *28-1573   *28-1574   *28-1582   *28-1583   *28-1584   *29-1620
                             *30-1649   *32-1713    *36-1830   *36-1832   *37-1870   *37-1872   *38-1912   *39-1990   *39-1993
                             *39-2015   *39-2020    *39-2040   *39-2044   *39-2046   *40-2133   *40-2136   *40-2157   *40-2161
                             *41-2273   *42-2338    *42-2339   *42-2340   *42-2344   *42-2345   *42-2346   *42-2389   *42-2390
                             *42-2391   *42-2419    *42-2423   *42-2426   *42-2427   *42-2428   *45-2583   *45-2584   *45-2593
                             *49-2774   *50-2831    *53-3019   *53-3020   *53-3022   *59-3341   *60-3379   *61-3433   *62-3503
                             *62-3515   *62-3520    *62-3544   *62-3575   *62-3616   *62-3620   *62-3643   *63-3672   *63-3677
                             *63-3708   *74-4009    *74-4025   *74-4028   *74-4065   *75-4094   *75-4103   *75-4123   *76-4139
                             *76-4142   *77-4164    *78-4179   *81-4271   *81-4275   *82-4321   *82-4328   *82-4347   *84-4415
S.COST   = ******    GX      *17-797
S.CTL      000006            #5-82
S.DLCF     000102             20-927
S.ERR      000010            #5-82
S.FLG    = ******    GX       11-487     11-500     *12-545    *12-553     12-558     16-733    *23-1071   *23-1081   *24-1111
                              24-1116   *24-1136     24-1140   *33-1748   *34-1776
S.LBE      000044            #5-82       21-984
S.LEN    = ******    GX       12-563     24-1120     24-1144
S.LGTH   = 000050            #5-82
S.LIN      000004            #5-82
S.LINK     177776            #5-82       5-82
S.LSA      000005            #5-82       21-985
S.NMST   = ******    GX      *23-1072   *24-1112
S.OWNR   = ******    GX      *16-737     81-4263
S.PLA      000010            *14-640
```

```
                                        ;+
                                        ; $VEXP - FRONT END FOR BINARY TO REAL PROCESSING
                                        ;
                                        ; THIS ROUTINE CALLS $VEX2 IF NO ERRORS HAVE OCCURRED DURING TEMPLATE
                                        ; PROCESSING AND $VEX3 OTHERWISE.
                                        ;
                                        ; INPUTS:
                                        ;       $ADDR=ADDRESS OF WHERE REAL DATA GOES
                                        ;       $NALL=NUMBER OF ALLOCATIONS AND LIBRARIES THIS TEMPLATE
                                        ;       $FLAGS INDICATES WHETHER PROCESS IS DLC ONLY OR NOT (MAPPED ONLY)
                                        ;       F1.ERR SET IF ANY ERRORS OCCURRED
                                        ;
                                        ; OUTPUTS:
                                        ;       ALL REGISTERS=DESTROYED
                                        ;-
000000                                  $VEXP::
000000  022767  177777  000000G                 CMP     #-1,$ADDR               ;IS IT EMPTY LLC DATA BASE?
000006  001422                                  BEQ     30$                     ;BR IF YES
000010  005767  000000G                         TST     $FLAG1                  ;DID AN ERROR OCCUR?
000014  100407                                  BMI     20$                     ;BR IF YES
000016  032767  000000G 000000G                 BIT     #FL.CHA,$FLAGS          ; MUX CHARACTERISTICS UPDATE ONLY ?
000024  001011                                  BNE     25$                     ; YES ..
000026                                  10$:    CALL    $VEX2                   ;EXPAND THE LINE TABLE
000032  000410                                  BR      30$                     ;AND LEAVE

000034  005767  000000G                 20$:    TST     $NALL                   ;ANYTHING TO CLEAN UP?
000040  001405                                  BEQ     30$                     ;BR IF NO
000042                                          CALL    $VEX3                   ;DO THE CLEANUP
000046  000402                                          BR      30$

000050                                  25$:    CALL    $VEX5                   ; UPDATE THE LINE CHARACTERISTICS

000054                                  30$:    RETURN


        000001                                  .END
```

```
000560  156001  000000G                  BISB    $SLN(R0),R1         ;       ...
000564  010124                           MOV     R1,(R4)+            ; STORE IT
000566                                   RETURN

                                ;
                                ; LOCAL NODE ADDRESS
                                ;

000570                          NOD:
000570                                   GETRV   $DECPT,#D$END       ; READ DECNET HOME BLOCK

000610  016724  000014G                  MOV     $BFR+D$LNUM,(R4)+ ; COPY LOCAL NODE ADDRESS
000614                                   RETURN

                                ;
                                ; PROTOCOL OVERHEAD SIZE
                                ;

000616  112524                  PRTOV:   MOVB    (R5)+,(R4)+         ; COPY PROTOCOL OVERHEAD SIZE
000620                                   RETURN


                                ;
                                ; STATION NUMBER (BYTE)
                                ;
000622  105767  000450'         STNB:    TSTB    FLAGS               ; IS IT MTP EXPANSION?
000626  100404                           BMI     10$                 ; IF MI, YES
000630  022767  000001  000000G          CMP     #1,$$MTP            ; ARE WE DEALING WITH ONLY 1 STATION ?
000636  001005                           BNE     101$                ; IF NE, NO
000640  016700  000446'         10$:     MOV     INDEX,R0            ; GET CURRENT STATION INDEX
000644  116024  000000G                  MOVB    $$SNUM(R0),(R4)+ ; STORE STATION NUMBER
000650                                   RETURN
000652  112724  177777          101$:    MOVB    #-1,(R4)+           ; DLC STNB NOT IN MPTAB FILE
000656                                   RETURN

                                ;
                                ; STATION NUMBER (WORD)
                                ;
000660  105767  000450'         STNW:    TSTB    FLAGS               ; IS IT MTP EXPANSION?
000664  100404                           BMI     10$                 ; IF MI, YES
000666  022767  000001  000000G          CMP     #1,$$MTP            ; ARE WE DEALING WITH ONLY 1 STATION ?
000674  001007                           BNE     101$                ; IF NE, NO
000676  016700  000446'         10$:     MOV     INDEX,R0            ; GET CURRENT STATION INDEX
000702  005001                           CLR     R1                  ; GET STATION NUMBER
000704  156001  000000G                  BISB    $$SNUM(R0),R1       ;       ...
000710  010124                           MOV     R1,(R4)+            ; STORE IT
000712                                   RETURN
000714  012724  177777          101$:    MOV     #-1,(R4)+           ; DLC STNW NOT IN MPTAB FILE
000720                                   RETURN

                                ;
                                ; UNIT NUMBER (BYTE)
                                ;
000722  005767  000450'         UNB:     TST     FLAGS               ; IS IT EITHER MUX OR MTP EXPANSION?
000726  001403                           BEQ     101$                ; IF EQ, NO
000730  116724  000446'                  MOVB    INDEX,(R4)+         ; STORE CURRENT MUX INDEX
000734                                   RETURN
```

```
Symbol table

ADDRB   000062R       D$$L11= 000001            LN.LOO= 000003            PR7   = ****** GX         X3CH11  002334R
ADDRW   000102R       D$$YNC= 000000            LN.OAU= 000003            P$$P45= 000000            $ADDR = ****** GX
APR     002212R       D$$YNM= 000000            LN.OFF= 000001            P$$WRD= 000000            $BFR  = ****** GX
A$$CHK= 000000        EOF     000356R           LN.ON = 000000            Q$$OPT= 000010            $BIAS = ****** GX
A$$CPS= 000000        E$$XPR= 000000            LN.OOP= 000004            R$$DER= 000000            $BIN  = ****** GX
A$$PRI= 000000        FILE    001212R           LN.OPE= 000001            R$$K11= 000001            $BUFCK= ****** GX
A$$TRP= 000000        FLAGA   000004R       002 LN.REF= 000002            R$$SND= 000000            $CTIM = ****** GX
BLKB    000130R       FLAGS   000450R       002 LN.SER= 000002            R$$11M= 000000            $DFCPT= ****** GX
BLKW    000142R       FL.DDM= ****** GX         LN.STA= 000017            SAVALC  001102R           $FLAGS= ****** GX
BYTE    000154R       FL.KMX= ****** GX         LN.SUB= 000360            SCOM    001064R           $GETLB= ****** GX
CORE    000774R       FL.MUX= ****** GX         LN.TRI= 000006            SECSR   000446R           $GETRV= ****** GX
CSR     000160R       F$$LVL= 000001            LSTHD   000424R           SF.ACT= 000200            $HMBUF= ****** GX
CTIM    000172R       F.NRBD= ****** GX         LTAB    000440R           SF.ENA= 000100            $ICBAD= ****** GX
C$$CKP= 000000        F.PRLT= 000001            L$$ASG= 000000            SF.LPB= 000004            $ICBLN= ****** GX
C$$ORE= 000400        F.RSIZ= ****** GX         L$$DRV= 000000            SF.MFL= 000040            $ICBSZ= ****** GX
C$$RSH= 177564        F.URBD= ****** GX         L$$P11= 000001            SF.PAC= 000020            $LDA  = ****** GX
DPRB    002224R       F.WRIT= 000002            L$$11R= 000000            SF.REA= 000010            $LINK = ****** GX
DPRW    002232R       G$$TPP= 000000            L.COST  000015            SF.SER= 000001            $MXADD= ****** GX
DVCHA   000200R       G$$TSS= 000000            L.CTL   000012            SF.SVC= 000002            $MXSIZ= ****** GX
D$AMXC  000072        G$$TTK= 000000            L.CVA   177776            SF.UNL= 000040            $PTR  = ****** GX
D$AMXH  000074        G$$WRD= 000000            L.DDM   000002            SLNB    000514R           $PUTRC= ****** GX
D$ANN   000000        ICBL    000212R       002 L.DDS   000004            SLNW    000540R           $RADDR= ****** GX
D$BRPR  000102        ICBLK   000006R       002 L.DLC   000003            STNB    000622R           $SADDR= ****** GX
D$BRTM  000100        ICBPT   000002R       002 L.DLM   000006            STNW    000660R           $SLN  = ****** GX
D$DELF  000045        INDEX   000446R       002 L.DLS   000010            SYSFDB= ****** GX         $SLTA = ****** GX
D$DELW  000046        INT     001122R           L.FLG   000000            S$$WRG= 000000            $TMPBF= ****** GX
D$END = 000104        I$$RAR= 000000            L.KRBA  000016            S$$YSZ= 007600            $UDA  = ****** GX
D$FNB   000034        I$$RDN= 000000            L.LEN = 000022            S.COST  000001            $VECT1= ****** GX
D$HIOR  000024        JSROF = 000012            L.MPF   000022            S.FLG   000000            $VEX2   000000RG
D$HOST  000022        KRBB    000214R       002 L.NMST  000020            S.LEN   000004            $X2HLD= ****** GX
D$INAC  000044        K$$CNT= 177546            L.NSTA  000014            S.NMST  000002            $X2MBS= ****** GX
D$INCT  000042        K$$CSR= 177546            L.OWNR  000021            S.OWNR  000003            $X2MRT= ****** GX
D$IPL   000051        K$$LDC= 000000            L.UNT   000013            TABLE   000274R       002 $X2MWS= ****** GX
D$LID   000020        K$$TPS= 000074            MPTAB   001276R           TIME    002206R           $X2RET= ****** GX
D$LNAM  000006        LB.ADR= ****** GX         MXPTB   001754R           T$$KMG= 000000            $X3CAL= ****** GX
D$LNUM  000014        LD$LP = 000000            MXTAB   001204R           T$$MIN= 000000            $X3CLR= ****** GX
D$LST   000047        LFILE   001304R           M$$CRB= 000124            UNB     000722R           $X3DBS= ****** GX
D$MAXC  000064        LF.ACT= 100000            M$$CRX= 000000            UNW     000744R           $X3DWS= ****** GX
D$MAXH  000066        LF.BRO= 000400            M$$FCS= 000000            V$$LTR= 001000            $X3MBS= ****** GX
D$MAXV  000070        LF.BWT= 000007            M$$MGE= 000000            WORD    000766R           $X3MCL= ****** GX
D$MLL   000040        LF.ENA= 002000            M$$NET= 000000            WRVBL   001714R           $X3MRE= ****** GX
D$MNOD  000041        LF.LPB= 001000            M$$OVR= 000000            X$$DBT= 000000            $X3MRS= ****** GX
D$NA    000062        LF.MDC= 000100            NOD     000570R           X2CH01  002342R           $X3MWS= ****** GX
D$NBEA  000056        LF.MFL= 004000            NXTLN   000000R       002 X2CH02  002350R           $X3RES= ****** GX
D$NBRA  000054        LF.MTP= 000020            N$$ACC= 000001            X2CH03  002356R           $X3RST= ****** GX
D$NEND= 000054        LF.PAC= 000200            N$$BUF= 000001            X2CH04  002364R           $$APR = ****** GX
D$NLN   000030        LF.RDY= 040000            N$$LDV= 000001            X2CH05  002372R           $$CSR = ****** GX
D$NN    000060        LF.REA= 010000            N$$MCP= 000001            X3CH01  002240R           $$DCHA= ****** GX
D$OUTT  000043        LF.SER= 000040            N$$MLL= 000001            X3CH02  002246R           $$DPR = ****** GX
D$RETF  000050        LF.TIM= 000010            N$$MOV= 000010            X3CH03  002254R           $$LSAD= ****** GX
D$RNN   000002        LF.UNL= 020000            N$$NCT= 000001            X3CH04  002262R           $$MTP = ****** GX
D$RTMR  000076        LF.X2P= 000000            N$$PEM= 000001            X3CH05  002270R           $$PCHA= ****** GX
D$SEG   000036        LIBR    001434R           PECHA   000310R           X3CH06  002276R           $$PRI = ****** GX
D$SER   000032        LINKS   000360R           PRI     001172R           X3CH07  002304R           $$[illegible]SR= ****** GX
D$SQRL  000052        LN.CLO= 000000            PRINT   001570R           X3CH08  002312R           $$SNUM= ****** GX
D$$BUG= 177514        LN.DUM= 000005            PRLT    001516R           X3CH09  002320R           .EOF  = ****** GX
D$$ISK= 000000        LN.LOA= 000004            PRTOV   000616R           X3CH10  002326R           .MGMGE= ****** GX
```

```
                                             .SBTTL  CLEANUP ROUTINES
                                    ;***
                                    ; CLEANUP ROUTINES
                                    ;***

                                    ;
                                    ; DISCARD 3 BYTES
                                    ;
 000036  062705  000003             POP3:    ADD     #3,R5
 000042                                      RETURN

                                    ;
                                    ; DISCARD 2 BYTES
                                    ;
 000044  122525                     POP2:    CMPB    (R5)+,(R5)+
 000046                                      RETURN

                                    ;
                                    ; DISCARD 1 BYTE
                                    ;
 000050  005205                     POP1:    INC     R5
 000052                                      RETURN

                                    ;
                                    ; NO CLEANUP
                                    ;
 000054                             NULL:    RETURN
```

```
                                .SBTTL  EXPANSION ROUTINES
                        ;***
                        ;
                        ; EXPANSION ROUTINES
                        ;
                        ;       WITH THE EXCEPTION OF ''.DVCHA'' ''.PECHA'' AND ''.SECSR'', ALL PROCESSING
                        ;       CONSISTS OF UPDATING THE BINARY BUFFER POINTER (R5) AND THE LINE TABLE
                        ;       POINTER (R3).
                        ;
                        ;***

                        ;
                        ; LOCAL PHYSICAL STATION ADDRESS (BYTE)
                        ;
000102  005203          ADDRB:  INC     R3
000104                          RETURN
                        ;
                        ; LOCAL PHYSICAL STATION ADDRESS (WORD)
000106  005723          ADDRW:  TST     (R3)+
000110                          RETURN
                        ;
                        ; ZERO BYTE(S)
                        ;
000112  005000          BLKB:   CLR     R0              ; GET REPEAT COUNT
000114  152500                  BISB    (R5)+,R0        ;
000116  001002                  BNE     10$             ; ZERO BECOMES 256.
000120  012700  000400          MOV     #256.,R0        ;
000124  060003          10$:    ADD     R0,R3
000126                          RETURN

                        ;
                        ; ZERO WORD(S)
                        ;
000130  005000          BLKW:   CLR     R0              ; GET REPEAT COUNT
000132  152500                  BISB    (R5)+,R0        ;
000134  001002                  BNE     10$             ; ZERO BECOMES 256.
000136  012700  000400          MOV     #256.,R0        ;
000142  006300          10$:    ASL     R0              ; 2 BYTES PER WORD
000144  060003                  ADD     R0,R3
000146                          RETURN

                        ;
                        ; SINGLE BYTE
                        ;
000150  005205          BYTE:   INC     R5
000152  005203                  INC     R3
000154                          RETURN

                        ;
                        ; POINTER TO ALLOCATED CORE
                        ;
000156  005767  000000G CORE:   TST     .MGMGE          ; MAPPED TARGET SYSTEM ?
000162  100405                  BMI     10$             ; NO ..
000164  062705  000006          ADD     #6,R5
000170  062703  000004          ADD     #4,R3
000174  000404                  BR      20$
000176  062705  000004  10$:    ADD     #4,R5
```

```
VEX5        CREATED BY  MACRO  ON 29-JUN-85 AT 02:28      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE               REFERENCES

ADDRB       000102 R          5-83        #7-214
ADDRW       000106 R          5-84        #7-218
APR         001450 R          5-114       #9-602
BLKB        000112 R          5-85        #7-223
BLKW        000130 R          5-86        #7-233
BLOCK       000164 R         #5-153        10-625       10-657       10-661       10-671
BYTE        000150 R          5-87        #7-244
CORE        000156 R          5-88        #7-251
CSR         000210 R          5-89        #7-263
CTIM        000220 R          5-90        #7-270
CTRADD      000162 R         #5-152        6-184      *10-668
DPRB        001450 R          5-115       #9-601
DPRW        001454 R          5-116       #9-608
DVCHA       000224 R          5-91        #7-276
EOF         000504 R          5-92        #7-328
FILE        001050 R          5-93        #8-476
FLAGS       000154 R         #5-145       *6-175        7-276        7-305        7-402       *8-475        8-479       *8-497       *8-525
                             *9-531       *9-568
FLT         001460 R        #10-624
FLT16       001460 R          6-180      #10-623
FL.KMX    = ******   GX       7-407
F.RSIZ    = ******   GX      *7-290       *7-319       *7-423       *9-542
F.URBD    = ******   GX     *10-627      *10-673
INDEX       000152 R         #5-140       *6-174        7-278        7-307        7-404        8-481       *8-492        8-493       *8-496
                             *8-520        8-521       *8-524        9-551       *9-562        9-563       *9-567
INT         000506 R          5-94        #7-333
LFILE       001160 R          5-97        #8-509
LF.ACT    = 100000           #5-65
LF.BRO    = 000400           #5-65
LF.BWT    = 000007           #5-65
LF.ENA    = 002000           #5-65
LF.LPB    = 001000           #5-65
LF.MDC    = 000100           #5-65
LF.MFL    = 004000           #5-65
LF.MTP    = 000020           #5-65
LF.PAC    = 000200           #5-65
LF.RDY    = 040000           #5-65
LF.REA    = 010000           #5-65
LF.SER    = 000040           #5-65
LF.TIM    = 000010           #5-65
LF.UNL    = 020000           #5-65
LF.X2P    = 000000           #5-65
LIBR        000532 R          5-95        #7-344
LINKS       000564 R          5-96        #7-356
LN.CLO    = 000000           #5-65
LN.DUM    = 000005           #5-65
LN.LOA    = 000004           #5-65
LN.LOG    = 000003           #5-65
LN.OAU    = 000003           #5-65
LN.OFF    = 000001           #5-65
LN.ON     = 000000           #5-65
LN.OOP    = 000004           #5-65
```

```
    221                                                     .SBTTL  LINE-ID SUBEXPRESSION
    222
    223                                             ;
    224                                             ; LINE-ID SUBEXPRESSION
    225                                             ;
    226 000052                                          STATE$  LINID
    227 000052                                           TRAN$   $RAD50,,DEVNAM          ; DEVICE NAME
    228
    229 000052                                          STATE$
    230 000052                                           TRAN$   <'->
    231
    232 000052                                          STATE$
    233 000052                                           TRAN$   $NUMBR,,DEVCTL          ; CONTROLLER NUMBER
    234
    235 000052                                          STATE$
    236 000052                                           TRAN$   <'->,LINID1
    237 000052                                           TRAN$   $LAMDA,$EXIT
    238
    239 000052                                          STATE$  LINID1
    240 000052                                           TRAN$   $NUMBR,$EXIT,DEVUNT     ; UNIT NUMBER
    241
    242 000052                                          STATE$
```

```
VFGDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:28      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

LF.X2P   = 000000           #5-67
LINCTL     000044 R         #7-123      *18-440
LINNAM     000042 R         #7-122      *18-433
LINUNT     000045 R         #7-124      *18-447
LN.CLO   = 000000           #5-67
LN.DUM   = 000005           #5-67
LN.LOA   = 000004           #5-67
LN.LOO   = 000003           #5-67
LN.OAU   = 000003           #5-67
LN.OFF   = 000001           #5-67
LN.ON    = 000000           #5-67
LN.OOP   = 000004           #5-67
LN.OPE   = 000001           #5-67
LN.REF   = 000002           #5-67
LN.SER   = 000002           #5-67
LN.STA   = 000017           #5-67
LN.SUB   = 000360           #5-67
LN.TRI   = 000006           #5-67
L$DTEA     000020            16-392
L$LEN      000122            16-389
L.COST     000015           #5-67
L.CTL      000012           #5-67
L.CVA      177776           #5-67
L.DDM      000002           #5-67
L.DDS      000004           #5-67
L.DLC      000003           #5-67
L.DLM      000006           #5-67
L.DLS      000010           #5-67
L.FLG      000000           #5-67
L.KRBA     000016           #5-67
L.LEN    = 000022           #5-67
L.MPF      000022           #5-67
L.NMST     000020           #5-67
L.NSTA     000014           #5-67
L.OWNR     000021           #5-67
L.UNT      000013           #5-67
NEXT       000040 R         #7-121      *12-255      12-262      19-476      *19-477
NUMDTE   = 000020           #6-87
N$$VCT   = ******            16-388
PCKBCD     000756 R          12-278     #20-497
PF.XDF   = ******    GX      12-283
RTSPC    = ******    GX      8-128       8-129       8-130       8-131       8-132
R$R0     = 000002           #6-92
R$R1     = 000004           #6-93
R$R2     = 000006           #6-94
R$R3     = 000010           #6-95
R$R4     = 000012           #6-96       *16-400
R$R5     = 000014           #6-97
R$$EIS   = ******            20-508
R$$11D   = ******            20-508
SF.ACT   = 000200           #5-67
SF.ENA   = 000100           #5-67
```

```
    265
    266 000056                                          STATE$  PFLAGS
    267 000056                                           TRAN$    !END,$EXIT,PVEND
    268 000056                                           TRAN$    <',>
    269
    270 000056                                          STATE$
    271 000056                                           TRAN$    $NUMBR,END,PVFG         ; FLAGS BYTE
    272 000056                                           TRAN$    $LAMDA
    273
    274 000056                                          STATE$  END
    275 000056                                           TRAN$    $EOS,$EXIT
    276 000056                                           TRAN$    <';>,$EXIT
    277                                           ;
    278                                           ;     CHECK FOR DLM CIRCUIT NAME - SUBEXPRESSION
    279                                           ;
    280 000056                                          STATE$  CHKDLM
    281 000056                                           TRAN$    '-,,PVDLM
    282
    283 000056                                          STATE$
    284 000056                                           TRAN$    $DNUMB,,PVCTL
    285
    286 000056                                          STATE$
    287 000056                                           TRAN$    '.
    288
    289 000056                                          STATE$
    290 000056                                           TRAN$    $DNUMB,,PVSTA,PF.DLM,FLAG
    291
    292 000056                                          STATE$
    293 000056                                           TRAN$    <',>,$EXIT
    294
    295 000056                                          STATE$
```

```
VFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:29  Page 20-3
Symbol table

PF$STA= 000004        P$CHN   000004        SF.UNL= 000040        XA$URE= 000100        Z.NAM = ****** GX
PF$SVC= 040000        P$CNT   000005        SPACE = 000040        XF$NCO= 000020        $ALPHA= 000022
PF$UP = 000004        P$CTR   000034        SP$P4   000062RG  002 XF$OFF= 000200        $ALPOL= ****** GX
PF.DLM= 000001        P$DRCT  000015        STORID= ****** GX     XF$ORI= 000002        $ANY  = 000020
PF.PSV= ****** GX     P$DRTR  000036        SYNERR= ****** GX     XF$OTI= 000001        $BFR  = ****** GX
PORTNO  000004R   002 P$FLG   000012        SYSFDB= ****** GX     XF$TIL= 000010        $BLANK= 000006
PRTADD  000006R   002 P$FWD   000030        S$$WRG= 000000        XF$TIR= 000004        $CAT5 = ****** GX
PR$BED= 000200        P$ICCB  000046        S$$YSZ= 007600        XT$CHN= 000100        $CEACC= ****** GX
PR$BEU= 000100        P$IPL   000014        S.COST  000001        XT$DGR= 000002        $DIGIT= 000024
PR$BRD= 000040        P$LCD   000002        S.FLG   000000        XT$FAS= 000200        $DNUMB= 000014
PR$BRU= 000020        P$LEN   000052        S.LEN   000004        XT$INC= 000004        $EOS  = 000012
PR$DWN= 000002        P$LST   000000        S.NMST  000002        XT$OUC= 000010        $ERRN0  000064R   002
PR$LCC= 000010        P$NRNI  000040        S.OWNR  000003        XT$PVC= 000001        $ERRN1  000124R   002
PR$MOP= 000004        P$OCCB  000050        T$FLAG  000044        X$ABQ   000076        $ERRN2  000160R   002
PR$UP = 000001        P$PFQ   000006        T$LIF   000013        X$ALF   000055        $ERRN3  000212R   002
PSZ     000044R   003 P$PKSZ  000044        T$LIFL  000013        X$AUC   000056        $ERRO0  000312R   002
PS$CHR= 000016        P$RMX1  000016        T$LIFO  000013        X$CLEN= 000025        $ERRO5  000346R   002
PS$FAI= 000014        P$RMX2  000020        T$LIFS  000013        X$CTIM  000060        $ERRO9  000402R   002
PS$NTI= 000006        P$RPRI  000042        T$LIN   000000        X$DIAG  000003        $ERRP8  000256R   002
PS$OFF= 000000        P$RTIM  000003        T$LIPS  000006        X$DTE   000024        $ERR1T= ****** GX
PS$STR= 000002        P$STA1  000022        T$LLD   000012        X$FLG   000014        $EXIT = 000000
PS$UP = 000012        P$STA2  000023        T$LLDC  000045        X$GLEN  000106        $FAIL = 177777
PS$VER= 000010        P$TIM   000010        T$LLDL  000012        X$LCN   000026        $GETAD= ****** GX
PS$WT = 000004        P$TSCT  000026        T$LLDO  000012        X$LEN   000106        $GETRV= ****** GX
PT$BRO= 000200        P$TSIZ  000024        T$LLDS  000012        X$MOWN  000023        $LAMDA= 000000
PT$DRT= 000100        P$TYP   000001        T$LLEN  000046        X$NLRE  000052        $LLCTA= ****** GX
PT$END= 000004        P$$P45= 000000        T$LOPR  000002        X$NNRE  000054        $NUMBR= 000002
PT$LV1= 000002        P$$WRD= 000000        T$LTCL  000024        X$NPL   000006        $PDVNM= ****** GX
PT$LV2= 000001        Q$$OPT= 000010        T$LTIM  000026        X$NPR   000007        $PDVTA= ****** GX
PT$PH3= 000010        RF.CTL= 000003        T$LTPR  000014        X$NRBY  000032        $PFLAG= ****** GX
PT$XAR= 000020        RF.LD1= 000040        T$LTPS  000020        X$NRPK  000042        $PSIPT= ****** GX
PVCCTM  000012R   002 RF.LD2= 000100        T$NAPL  000004        X$NRRE  000053        $PUTAD= ****** GX
PVCDF   000000R   003 RF.TIM= 177400        T$NFE   000000        X$NTBY  000036        $QPVC   000000RG
PVCFLG  000035R   002 RF.TMO= 000400        T$NLEN  000010        X$NTPK  000046        $RADDR= ****** GX
PVCLCN  000010R   002 RF.WFC= 000200        T$NNUL  000002        X$PKSZ  000016        $RAD50= 000016
PVCNAM  000037R   002 RF.WTD= 000020        T$NOPL  000006        X$PR    000011        $RBLCK= ****** GX
PVCNML  000036R   002 RF.WTM= 000030        T$NRNI  000042        X$PRT   000021        $SAVAL= ****** GX
PVCOWN  000020R   002 RF.WTS= 000010        T$NRPL  000005        X$PS    000010        $SLTA = ****** GX
PVCPRT  000034R   002 RTSPC = ****** GX     T$NRUL  000007        X$RPR   000013        $SLTMA= ****** GX
PVCPSZ  000014R   002 R$R0  = 000002        T$NVR   000001        X$RPS   000012        $STRNG= 000004
PVCST   000000RG  003 R$R1  = 000004        T$RPRI  000040        X$RTRY  000002        $SUBXP= 000010
PVCT    000166R       R$R2  = 000006        T$SVC   000034        X$RXI   000104        $TRIB = ****** GX
PVCTL   001554R       R$R3  = 000010        T$T5    000030        X$RXQ   000072        $X3DBS= ****** GX
PVCWSZ  000016R   002 R$R4  = 000012        T$T6    000032        X$SS    000004        $X3DWS= ****** GX
PVDLM   001506R       R$R5  = 000014        T$$KMG= 000000        X$ST    000005        $$HTIM= ****** GX
PVEND   000346R       R$$DER= 000000        T$$MIN= 000000        X$TCLZ  000030        $$LTIM= ****** GX
PVFG    000314R       R$$K11= 000001        VF$RCV= 100000        X$TIMC  000001        $$$FLG= 177777
PVLCN   000130R       R$$SND= 000000        VF$XMT= 040000        X$TIMR  000000        $$$KEY= 000002
PVNAM   000056R       R$$11M= 000000        V$FLG   000000        X$TXI   000102        $$$R  = 000010
PVNAMX= 000006        SF.ACT= 000200        V$LEN   000022        X$TXQ   000066        $$$STA= 000000
PVOWN   000256R       SF.ENA= 000100        V$RCV   000002        X$TYP   000015        $$$TMP= 000013R   005
PVPSZ   000212R       SF.LPB= 000004        V$XMT   000012        X$USR   000022        .PNUMB= ****** GX
PVSTA   001626R       SF.MFL= 000040        V$$CTR= 001000        X$WAQ   000062        .PNUMH= ****** GX
PVWSZ   000234R       SF.PAC= 000020        WNDSZ   000066R   003 X$WSZ   000020        .PSTCN= ****** GX
PX$BLK= 000040        SF.REA= 000010        XA$NCO= 000020        X$$DBT= 000000        .PSTPT= ****** GX
PX$DLM= 000200        SF.SER= 000001        XA$RR = 000200        Z.DAT = ****** GX     .TPARS= ****** GX
PX$SVC= 000100        SF.SVC= 000002        XA$UCL= 000040        Z.MAP = ****** GX
```

```
    107                                         ;
    108                                         ; CONSTANT DEFINITIONS
    109                                         ;
    110          000007                                 MAXWND = 7                      ; MAXIMUM WINDOW SIZE
    111                                                 .SBTTL  LOCAL DATA
    112
    113                                         ;
    114                                         ; LOCAL DATA
    115                                         ;
    116 000000                                          .PSECT  DATA,D
    117 000000                                  NEXT:   .BLKW   1               ; ADDRESS TO STORE NEXT DTE DIGIT
    118 000002                                  HTIMR:  .BLKW   1               ; LOCAL COPY OF HELLO TIMER
    119 000004                                  LTIMR:  .BLKW   1               ; LOCAL COPY OF LISTEN TIMER
    120 000006                                  PKTSVC: .BLKW   1               ; PACKET SIZE
    121 000010                                  FLGSVC: .BLKW   1               ; FLAGS WORD
    122 000012                                  RTMSVC: .BLKW   1               ; RECALL TIMER
    123 000014                                  STASVC: .BLKW   1               ; STATION NUMBER
    124 000016                                  SLTA:   .BLKW   1               ; LOCAL COPY OF $SLTA
    125 000020                                  OWNSVC: .BLKB   1               ; OWNER
    126 000021                                  CTLSVC: .BLKB   1               ; CONTROLLER NUMBER
    127 000022                                  WNDSVC: .BLKB   1               ; WINDOW SIZE
    128 000023                                  RCLSVC: .BLKB   1               ; MAX RECALLS
    129 000024                                  COUNT:  .BLKB   1               ; NUMBER OF DIGITS IN DTE ADDRESS
    130                 000017                          DTELEN = 15.
    131 000025                                  DTESVC: .BLKB   DTELEN          ; DTE ADDRESS
    132                 000044'                         DTEEND = .
    133 000044                                  DTEPCK: .BLKB   <DTELEN+1>/2    ; DTE ADDRESS (BCD)
    134 000054                                  CVTBUF: .BLKB   1               ; CONVERSION BUFFER
    135                                                 .EVEN
```

```
                                             .SBTTL  FINFLT - FIND PLACE TO INSERT FILTER BLOCK

                                     ;+
                                     ;
                                     ; FINFLT - FIND PLACE TO INSERT FILTER BLOCK
                                     ;
                                     ; INPUTS:
                                     ;       R1 - FLAGS WORD FOR FILTER BLOCK TO BE INSERTED
                                     ;       R2 - EVENT CLASS NUMBER FOR FILTER BLOCK TO BE INSERTED
                                     ;       R3 - SINK NODE NUMBER FOR FILTER BLOCK TO BE INSERTED
                                     ;
                                     ; OUTPUTS:
                                     ;       R1 - UNMAPPED ADDRESS OF PREVIOUS BLOCK
                                     ;
                                     ;-

015340  012704  000000G              FINFLT: MOV     #$FILHD,R4              ; GET START OF FILTER BLOCKS IN LIST
015344  011467  000000G                      MOV     (R4),$BFR               ; GET FIRST BLOCK IN LIST
015350  010146                               MOV     R1,-(SP)                ; SAVE FLAGS WORD
015352  010467  000164'                      MOV     R4,CURUNM               ; INIT UNMAPPED ADDR OF CURRENT BLOCK
015356  016767  000164' 000166' 10$:         MOV     CURUNM,PRVUNM           ; SAVE UNMAPPED ADDR OF PREVIOUS BLOCK
015364  016767  000000G 000164'              MOV     $BFR,CURUNM             ; GET NEXT FILTER BLOCK IN LIST
015372  001454                               BEQ     60$                     ; BR IF NO MORE IN LIST
015374                                       CEACC$  CURUNM,R4               ; CONVERT TO MAPPED ADDRESS
015406                               20$:    GETAD   R4,#F.LEN               ; READ IN FILTER BLOCK
015424  016767  000002G 000000G              MOV     F.CLS+$BFR,$RSIZE       ; GET LENGTH OF BLOCK
015432  042767  177700  000000G              BIC     #^C<FF.MSK>,$RSIZE      ; ...
015440  022767  000016  000000G              CMP     #F.LEN,$RSIZE           ; IS THIS THE CORRECT LENGTH?
015446  001402                               BEQ     25$                     ; BR IF YES
015450                                       GETAD                           ; ELSE READ IN ENTIRE BLOCK
015454  032716  100000               25$:    BIT     #FF.REM,(SP)            ; IS SINK NODE = EXECUTOR?
015460  001005                               BNE     30$                     ; BR IF NO
015462  032767  100000  000014G              BIT     #FF.REM,F.FLG+$BFR      ; IS SINK NODE = EXECUTOR NODE?
015470  001412                               BEQ     50$                     ; BR IF YES - SORT BY CLASS
015472  000414                               BR      60$                     ; ELSE LINK BLOCK HERE

015474  032767  100000  000014G 30$:         BIT     #FF.REM,F.FLG+$BFR      ; IS SINK NODE = EXECUTOR NODE?
015502  001404                               BEQ     40$                     ; BR IF YES - CONTINUE SCAN
015504  020367  000026G                      CMP     R3,F.REM+$BFR           ; SORT BY SINK NODE
015510  103405                               BLO     60$                     ; IF LOWER, LINK BLOCK INTO LIST
015512  001401                               BEQ     50$                     ; IF EQ, SORT BY CLASS
015514  000720                       40$:    BR      10$

015516  020267  000002G              50$:    CMP     R2,F.CLS+$BFR           ; SORT BY CLASS
015522  103315                               BHIS    10$                     ; IF HIGHER OR SAME, CHECK NEXT BLOCK

015524  016701  000166'              60$:    MOV     PRVUNM,R1               ; RETRIEVE UNMAPPED ADDR OF PREVIOUS
015530  005726                               TST     (SP)+                   ; CLEAN UP STACK
015532                                       RETURN

```

```
                                                .SBTTL  STREVT - STORE THE EVENT MASKS

                                        ;+
                                        ;
                                        ; STREVT - STORE THE EVENT MASKS IN THE FILTER BLOCK
                                        ;
                                        ; INPUTS:
                                        ;       RQB+LO.EVT - EVENT FILTER MASKS TO STORE
                                        ;       R1 - ADDRESS OF EVENT FILTER MASKS IN FILTER BLOCK
                                        ;
                                        ; OUTPUTS:
                                        ;       R1 - ADDRESS OF NEXT WORD IN FILTER BLOCK
                                        ;
                                        ;-

017432  016721  000004G                 STREVT: MOV     RQB+LO.EVT+0,(R1)+      ; STORE EVENT FILTERS
017436  016721  000006G                         MOV     RQB+LO.EVT+2,(R1)+      ; ...
017442  016721  000010G                         MOV     RQB+LO.EVT+4,(R1)+      ; ...
017446  016721  000012G                         MOV     RQB+LO.EVT+6,(R1)+      ; ...
017452                                          RETURN
```

```
                                                  .SBTTL  FNDCTR - FIND COUNTER BLOCK

                                          ;+
                                          ;
                                          ; FNDCTR - FIND COUNTER BLOCK ENTRY FOR SPECIFIED CIRCUIT
                                          ;
                                          ; INPUTS:
                                          ;       $SLTA - SLT ADDRESS FOR CIRCUIT
                                          ;       $TRIB - TRIB NUMBER FOR CIRCUIT
                                          ;
                                          ; OUPUTS:
                                          ;       CARRY CLEAR:
                                          ;          $BFR - FOUND COUNTER BLOCK
                                          ;          $RADDR - ADDRESS OF COUNTER BLOCK
                                          ;          $RBLCK - APR BIAS FOR COUNTER BLOCK
                                          ;       CARRY SET:
                                          ;          COUNTER BLOCK NOT FOUND
                                          ;
                                          ;-
021022  016700  000000G                   FNDCTR: MOV     $SLTA,R0                ; GET THE SLT ADDRESS
021026  032760  000000G 000000G                   BIT     #LF.RDY,L.FLG(R0)       ; IS LINE LOADED?
021034  001474                                    BEQ     40$                     ; BR IF NO - PARAMETER NOT APPLICABLE
021036  116001  000000G                           MOVB    L.OWNR(R0),R1           ; ASSUME OWNER = LINE OWNER
021042  105760  000000G                           TSTB    L.NSTA(R0)              ; MULTIPOINT LINE ?
021046  001412                                    BEQ     10$                     ; IF EQ, NO
021050  010001                                    MOV     R0,R1                   ; GET SLT ADDRESS
021052  062701  000000G                           ADD     #L.MPF,R1               ; POINT TO STATION ADDRESS
021056  016746  000000G                           MOV     $TRIB,-(SP)             ; ...
021062  006316                                    ASL     (SP)                    ; ...
021064  006316                                    ASL     (SP)                    ; ...
021066  062601                                    ADD     (SP)+,R1                ; ...
021070  116101  000000G                           MOVB    S.OWNR(R1),R1           ; ELSE, GET TRIB OWNER
021074  066701  000000G                   10$:    ADD     $PDVTA,R1               ; POINT TO PDV TABLE ENTRY
021100  011101                                    MOV     (R1),R1                 ; GET THE PDV ADDRESS
021102  022761  114224  000000G                   CMP     #^RXPT,Z.NAM(R1)        ; IS THE OWNER XPT ?
021110  001046                                    BNE     40$                     ; IF NE, NO-PARAMETER NOT APPLICABLE
021112  016746  000000G                           MOV     $SLN,-(SP)              ; GET SLN NUMBER
021116  116766  000000G 000001                    MOVB    $TRIB,1(SP)             ; SET TRIB NUMBER ONTO HIGH BYTE
021124  012605                                    MOV     (SP)+,R5                ; RETRIEVE SLN/STA
021126                                            GETRV   Z.DAT(R1),#N$XLEN       ; POINT TO XPT'S DDB
021146  016767  000102G 000000G                   MOV     N$TLC+2+$BFR,$RBLCK     ; MAP TO CIRCUIT COUNTERS
021154  016700  000100G                           MOV     N$TLC+$BFR,R0           ; GET THE NUMBER OF LINE COUNTERS
021160  016701  000104G                           MOV     N$TLC+4+$BFR,R1         ; POINT TO THE FIRST
021164                                    20$:    GETAD   R1,#T$LLEN              ; READ IN COUNTER BLOCK
021202  005300                                    DEC     R0                      ; AT END OF COUNTERS ?
021204  002410                                    BLT     40$                     ; IF LT, YES-PARAMETER NOT APPLICABLE
021206  026705  000000G                           CMP     T$LIN+$BFR,R5           ; ARE THESE THE COUNTERS ?
021212  001403                                    BEQ     30$                     ; IF EQ, YES - READ THEM
021214  062701  000046                            ADD     #T$LLEN,R1              ; ELSE, POINT TO NEXT COUNTERS
021220  000761                                    BR      20$                     ; AND KEEP LOOKING
021222  000241                            30$:    CLC                             ; INDICATE FOUND IT
021224  000401                                    BR      50$                     ; AND EXIT
021226  000261                            40$:    SEC
021230                                    50$:    RETURN
```

```
SYMBOL CRO.S REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

CL$XL3  = 013600            #5-79        5-79         10-355
CL$X2S  = 013500            #5-79        5-79         10-352
CM$SET  = ******  GX         27-1381     28-1485      36-1834      37-1874      41-2198      42-2302      44-2494      47-2683      57-3185
                             58-3257     58-3268      58-3277      59-3323      62-3482      78-4181
CNODT5    003714 R           25-1175    #25-1243
CONNAM    000160 R          #6-158       49-2770     *58-3259     *58-3260      58-3262      58-3263
CURUNM    000164 R          #6-159      *27-1447     *28-1556     *29-1615      29-1616     *29-1617      29-1619     *30-1643      30-1644
                            *30-1645     30-1648     *41-2261     *42-2386     *59-3318     *60-3374      60-3375     *60-3376      60-3378
                            *61-3427     61-3428     *61-3430      61-3432      61-3459
CUSTEV  = 000740            #8-211       46-2643
C.LGTH  = 000020             6-133       82-4305      82-4328      82-4354
C.RQT     000002            *82-4317
C.RSI     000012            *82-4318    *82-4319
C.SSHT  = 000004             82-4317
C.TCB     000004            *82-4320     82-4334      82-4334
C.TIM     000006            *82-4316    *82-4322     *82-4323      82-4336      82-4336      82-4339      82-4339
DEAFIL    017600 R           57-3223     62-3534      62-3609      62-3632     #72-3929
DEFCON    000044 R          #6-130       58-3259      58-3260
DEFDEV    000032 R          #6-125       50-2819
DEFFIL    000020 R          #6-122       6-128        6-132        50-2804      58-3246      58-3279
DEFMON    000040 R          #6-129       58-3270      58-3271
DEFTYP    000026 R          #6-123
DEFUIC    000036 R          #6-127
DEFUNT    000034 R          #6-126       50-2820
DEFVER    000030 R          #6-124
DF.CRC  = 000010            #5-82
DF.FOC  = 000020            #5-82
DF.LEN  = 000030            #6-132       58-3249
DF.RAN  = 000040            #5-82
DF.UNK  = 000100            #5-82
DISMNT    020330 R           39-2052     39-2068     #75-4088
DLXPX     000125 R          #6-146      *11-399      *11-403       11-428
DL$AST  = 000002            #5-79
DL$HLT  = 000000            #5-79
DL$IST  = 000001            #5-79
DL$MAI  = 000004            #5-79
DL$OFF  = 000001            #5-79
DL$ON   = 000000            #5-79
DL$RUN  = 000003            #5-79
DL$SHU  = 000002            #5-79
DL$SYN  = 000005            #5-79
DSBLIN    003230 R           11-502     #24-1097
DV.F11  = 040000             50-2836
DV.REC  = 000001             49-2779
D$ANN     000000            *42-2342     42-2418     *42-2424
D$END   = 000104             26-1267     26-1290      26-1307      26-1324      26-1341      26-1358      27-1399      27-1407      28-1573
                             28-1583     42-2339      42-2345      42-2390      42-2427      53-3020      74-4009      74-4025      78-4179
D$HOST    000022            *78-4183    *78-4186
D$INAC    000044            *26-1342
D$INCT    000042            *26-1308
D$LNAM    000006             26-1271     26-1273      26-1275
D$LNUM    000014             26-1269     53-3021      78-4186
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 15
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL  VALUE          REFERENCES

S.PSA     000003       #5-82       *13-602
S.RBE     000042       #5-82
S.RCB     000020       #5-82
S.RCV     000014       #5-82
S.RDE     000034       #5-82
S.RTY     000001       #5-82
S.STA     000000       #5-82
S.STLN    000035        20-928
S.STPN    000034       *13-597
S.TLZ     000012       #5-82
S.TMO     000040       #5-82
S.XDE     000036       #5-82
S.XID     000002       #5-82
S.XMB     000030       #5-82
S.XMP     000024       #5-82
TF.FFE  = 000004       #5-82
TF.ROF  = 000001       #5-82
TF.RTO  = 000002       #5-82
TF.TAE  = 000010       #5-82
TF.TOF  = 000004       #5-82
TF.TTO  = 000010       #5-82
TF.URE  = 000001       #5-82
TF.USA  = 000002       #5-82
TICKS   = 000001       #6-134       39-2016     40-2158
TMPMSK    000076 R     #6-139       57-3178     57-3202     57-3212     62-3553     62-3559     62-3565     62-3568     62-3606
                        62-3612     62-3624
TR.ALF    000102       #5-82
TR.DEV    000100       #5-82
TR.LEN  = 000210       #5-82
TR.MST    000074       #5-82
TR.PRE    000076       #5-82
TR.TLZ    000072       #5-82
TSKSCH    021630 R      39-1953     39-1966     39-2060     40-2098     40-2113     40-2174     75-4090    #84-4414
TX.NXM  = 000002       #5-82
TX.OVR  = 000001       #5-82
TX.TMO  = 000004       #5-82
TY$CHA  = 000400       #5-73
TY$EVE  = 001000       #5-73
TY$STA  = 002000       #5-73
TY$SUM  = 000000       #5-73
T$FLAG    000044       #5-68
T$LIF     000013       #5-68
T$LIFL    000013       #5-68
T$LIFO    000013       #5-68
T$LIFS    000013       #5-68
T$LIN     000000       #5-68        81-4278
T$LIPS    000006       #5-68
T$LLD     000012       #5-68
T$LLDC    000045       #5-68
T$LLDL    000012       #5-68
T$LLDO    000012       #5-68
T$LLDS    000012       #5-68
```

```
VEXP - VNP EXPAND TEMPLATE PROC MACRO V05.03b Saturday 29-Jun-85 02:27  Page 5-1
Symbol table

A$$CHK= 000000        FL.CHA= ****** GX     L$$ASG= 000000        N$$MCP= 000001        S$$YSZ= 007600
A$$CPS= 000000        F$$LVL= 000001        L$$DRV= 000000        N$$MLL= 000001        T$$KMG= 000000
A$$PRI= 000000        G$$TPP= 000000        L$$P11= 000001        N$$MOV= 000010        T$$MIN= 000000
A$$TRP= 000000        G$$TSS= 000000        L$$11R= 000000        N$$NCT= 000001        V$$CTR= 001000
C$$CKP= 000000        G$$TTK= 000000        M$$CRB= 000124        N$$PEM= 000001        X$$DBT= 000000
C$$ORE= 000400        G$$WRD= 000000        M$$CRX= 000000        P$$P45= 000000        $ADDR = ****** GX
C$$RSH= 177564        I$$RAR= 000000        M$$FCS= 000000        P$$WRD= 000000        $FLAGS= ****** GX
D$$BUG= 177514        I$$RDN= 000000        M$$MGE= 000000        Q$$OPT= 000010        $FLAG1= ****** GX
D$$ISK= 000000        K$$CNT= 177546        M$$NET= 000000        R$$DER= 000000        $NALL = ****** GX
D$$L11= 000001        K$$CSR= 177546        M$$OVR= 000000        R$$K11= 000001        $VEXP   000000RG
D$$YNC= 000000        K$$LDC= 000000        N$$ACC= 000001        R$$SND= 000000        $VEX2 = ****** GX
D$$YNM= 000000        K$$TPS= 000074        N$$BUF= 000001        R$$11M= 000000        $VEX3 = ****** GX
E$$XPR= 000000        LD$LP = 000000        N$$LDV= 000001        S$$WRG= 000000        $VEX5 = ****** GX


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000056    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8920 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.03
SY:VEXP.V2,[132,134]VEXP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VEXP
```

```
    421 000736  112724  177777          101$:   MOVB    #-1,(R4)+       ; UNB NOT IN EITHER MXTAB OR MPTAB FILE
    422 000742                                  RETURN
    423
    424                                 ;
    425                                 ; UNIT NUMBER (WORD)
    426                                 ;
    427 000744  005767  000450'         UNW:    TST     FLAGS           ; IS IT EITHER MUX OR MTP EXPANSION?
    428 000750  001403                          BEQ     101$            ; IF EQ, NO
    429 000752  016724  000446'                 MOV     INDEX,(R4)+     ; STORE CURRENT MUX INDEX
    430 000756                                  RETURN
    431 000760  012724  177777          101$:   MOV     #-1,(R4)+       ; UNW NOT IN EITHER MXTAB OR MPTAB FILE
    432 000764                                  RETURN
    433
    434                                 ;
    435                                 ; SINGLE WORD
    436                                 ;
    437 000766  112524                  WORD:   MOVB    (R5)+,(R4)+     ; COPY A WORD
    438 000770  112524                          MOVB    (R5)+,(R4)+
    439 000772                                  RETURN
```

```
Symbol table


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        002400    001   (RW,I,LCL,REL,CON)
DATA    000452    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12633 Words  ( 50 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:22.10
SY:VEX2.V2,[132,134]VEX2/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VEX2
```

```
    188                                                     ;***
    189                                                     ; DE-ALLOCATION CLEANUP ROUTINES
    190                                                     ;***
    191
    192                                                     ;
    193                                                     ; SINGLEWORD ADDRESSABLE STORAGE
    194                                                     ;
    195 000056  112546                                      XSCOM:  MOVB    (R5)+,-(SP)     ; ASSEMBLE ADDRESS
    196 000060  112566  000001                                      MOVB    (R5)+,1(SP)     ;    ...
    197 000064  012600                                              MOV     (SP)+,R0        ;    ...
    198 000066  112546                                              MOVB    (R5)+,-(SP)     ; AND COUNT
    199 000070  112566  000001                                      MOVB    (R5)+,1(SP)     ;    ...
    200 000074  012601                                              MOV     (SP)+,R1        ;    ...
    201 000076  001402                                              BEQ     10$             ; IF ZERO, DO NOTHING
    202 000100                                                      CALL    $DEA16          ; DEALLOCATE SINGLEWORD STORAGE
    203 000104                                              10$:    RETURN
    204
    205                                                     ;
    206                                                     ; DOUBLEWORD ADDRESSABLE STORAGE
    207                                                     ;
    208 000106  112546                                      XCORE:  MOVB    (R5)+,-(SP)     ; ASSEMBLE BIAS
    209 000110  112566  000001                                      MOVB    (R5)+,1(SP)     ;    ...
    210 000114  012600                                              MOV     (SP)+,R0        ;    ...
    211
    212 000116  005767  000000G                                     TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM ?
    213 000122  100401                                              BMI     5$              ; NO
    214
    215 000124  122525                                              CMPB    (R5)+,(R5)+     ; DISCARD DISPLACEMENT
    216
    217 000126  112546                                      5$:     MOVB    (R5)+,-(SP)     ; ASSEMBLE COUNT
    218 000130  112566  000001                                      MOVB    (R5)+,1(SP)     ;    ...
    219 000134  012601                                              MOV     (SP)+,R1        ;    ...
    220 000136  001402                                              BEQ     10$             ; IF ZERO, DO NOTHING
    221 000140                                                      CALL    $DEA18          ; DEALLOCATE DOUBLEWORD STORAGE
    222 000144                                              10$:    RETURN
```

```
000202  062703  000002                          ADD     #2,R3
000206                                  20$:    RETURN


                                        ;
                                        ; CSR
                                        ;
000210  005205                          CSR:    INC     R5
000212  062703  000002                          ADD     #2,R3
000216                                          RETURN

                                        ;
                                        ; COUNTER TIMER
                                        ;
000220  005723                          CTIM:   TST     (R3)+
000222                                          RETURN

                                        ;
                                        ; DEVICE CHARACTERISTICS
                                        ;
000224  005767  000154'                 DVCHA:  TST     FLAGS             ; IS IT MUX EXPANSION ?
000230  001454                                  BEQ     10$               ; NO ..
000232  126467  000013  000152'                 CMPB    L.UNT(R4),INDEX   ; DO UNIT NUMBERS MATCH ?
000240  001050                                  BNE     10$               ; NO ..
000242  010377  000000G                         MOV     R3,@$ADRZ         ; VIRTUAL ADDRESS
000246  005077  000000G                         CLR     @$ADRX            ; BIAS
000252  005767  000000G                         TST     .MGMGE            ; MEMORY MANAGEMENT SYSTEM?
000256  100413                                  BMI     5$                ; NO IF MINUS
000260  020327  120000                          CMP     R3,#120000        ; PARTITION LINE TABLE
000264  103410                                  BLO     5$                ; NO IF LO
000266  010346                                  MOV     R3,-(SP)          ; LINE TABLE ADDRESS
000270  042716  160000                          BIC     #160000,(SP)      ; CLEAR THE APR BITS
000274  012677  000000G                         MOV     (SP)+,@$ADRZ      ; NEW VIRTUAL ADDRESS
000300  016777  000000G 000000G                 MOV     $BIAS,@$ADRX      ; BIAS OF PROCESS
000306                                  5$:     GETAD   ,#4               ; ACCESS LINE TABLE
000320                                          SAVRG   R0                ; SAVE REG
000322  016700  000000G                         MOV     $SLTA,R0          ; COPY SLT ADDRESS
000326  010360  177776                          MOV     R3,L.CVA(R0)      ; COPY DEV CHAR POINTER
000332  066760  000000G 177776                  ADD     $LINK,L.CVA(R0)   ; ...
000340                                          RESRG   R0                ; RESTORE REG
000342  016767  000000G 000000G                 MOV     $$DCHA,$BFR       ; COPY DEVICE CHARACTERISTICS
000350  016767  000002G 000002G                 MOV     $$DCHA+2,$BFR+2   ; TWO WORDS
000356                                          PUTAD                     ; RE-WRITE LINE TABLE
000362  062703  000004                  10$:    ADD     #4,R3
000366                                          RETURN

                                        ;
                                        ; PROTOCOL EMULATOR CHARACTERISTICS
                                        ;
'00370  005767  000154'                 PECHA:  TST     FLAGS             ; IS IT MUX EXPANSION ?
 00374  001440                                  BEQ     10$               ; NO ..
 )0376  126467  000013  000152'                 CMPB    L.UNT(R4),INDEX   ; DO UNIT NUMBERS MATCH ?
 00404  001034                                  BNE     10$               ; NO ..
 00406  010377  000000G                         MOV     R3,@$ADRZ         ; VIRTUAL ADDRESS
'00412  005077  000000G                         CLR     @$ADRX            ; BIAS
 00416  005767  000000G                         TST     .MGMGE            ; MEMORY MANAGEMENT SYSTEM?
,00422  100413                                  BMI     5$                ; NO IF MINUS
000424  020327  120000                          CMP     R3,#120000        ; PARTITION LINE TABLE
```

```
VEX5      CREATED BY  MACRO  ON 29-JUN-85 AT 02:28      PAGE 2
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE             REFERENCES

LN.OPE  = 000001           #5-65
LN.REF  = 000002           #5-65
LN.SER  = 000002           #5-65
LN.STA  = 000017           #5-65
LN.SUB  = 000360           #5-65
LN.TRI  = 000006           #5-65
LSTHD     000572 R          5-98     #7-362
LTAB      000600 R          5-99     #7-368
L.COST    000015           #5-65
L.CTL     000012           #5-65
L.CVA     177776           #5-65     *7-293    *7-294
L.DDM     000002           #5-65      6-179
L.DDS     000004           #5-65      8-483     9-553
L.DLC     000003           #5-65
L.DLM     000006           #5-65
L.DLS     000010           #5-65
L.FLG     000000           #5-65
L.KRBA    000016           #5-65
L.LEN   = 000022           #5-65
L.MPF     000022           #5-65
L.NMST    000020           #5-65
L.NSTA    000014           #5-65
L.OWNR    000021           #5-65
L.UNT     000013           #5-65      7-278     7-307     7-405     8-481     9-551
MPTAB     001156 R          5-100    #8-508
MXPTB     001242 R          5-118    #9-531
MXTAB     001044 R          5-101    #8-475
NOD       000614 R          5-102    #7-381
PECHA     000370 R          5-119    #7-305
PRI       000606 R          5-103    #7-374
PRTOV     000620 R          5-104    #7-388
SCOM      000626 R          5-105    #7-395
SECSR     000640 R          5-106    #7-402
SF.ACT  = 000200           #5-65
SF.ENA  = 000100           #5-65
SF.LPB  = 000004           #5-65
SF.MFL  = 000040           #5-65
SF.PAC  = 000020           #5-65
SF.REA  = 000010           #5-65
SF.SER  = 000001           #5-65
SF.SVC  = 000002           #5-65
SF.UNL  = 000040           #5-65
SLNB      000776 R          5-107    #7-432
SLNW      001002 R          5-108    #7-438
STNB      001010 R          5-109    #7-444
STNW      001014 R          5-110    #7-450
SYSFDB  = ******   GX      *7-290    *7-319    *7-423    *9-542    *10-627   *10-673
S.COST    000001           #5-65
S.FLG     000000           #5-65
S.LEN     000004           #5-65
S.NMST    000002           #5-65
S.OWNR    000003           #5-65
```

```
                                                .SBTTL  DTE$DF ACTION ROUTINES
                                        ;
                                        ; START OF DTE ADDRESS
                                        ;
 000052  005067  000000G                DTEAST: CLR     DTEFLG                  ; INITIALISE FLAGS (ON/OFF BITS)
 000056  012700  000017                         MOV     #DTEALN,R0              ; GET LENGTH OF DTE ADDRESS
 000062  012701  000000'                        MOV     #DTEADD,R1              ; GET ADDRESS OF BUFFER
 000066  105021                         10$:    CLRB    (R1)+                   ; INITIALIZE BUFFER
 000070  005300                                 DEC     R0                      ; MORE TO INITIALIZE?
 000072  003375                                 BGT     10$                     ; BR IF YES
 000074  105067  000035'                        CLRB    COUNT                   ; INITIALIZE DTE DIGIT COUNT
 000100  012767  000000' 000040'                MOV     #DTEADD,NEXT            ; START STORING AT BEGINNING OF BUFFER
 000106                                         CALL    STRNXT                  ; STORE CHARACTER
 000112                                         RETURN


                                        ;
                                        ; PROCESS DTE ADDRESS DIGIT
                                        ;
 000114  026727  000040' 000016'        DTEDIG: CMP     NEXT,#DTEND             ; IS ADDRESS TOO MANY CHARACTERS?
 000122  101003                                 BHI     101$                    ; BR IF YES
 000124                                         CALL    STRNXT                  ; STORE NEXT DIGIT
 000130                                         RETURN
                                        ;
                                        ; ERRORS
                                        ;
 000132                                 101$:   EMSG$R  03                      ; ILLEGAL DTE ADDRESS


                                        ;
                                        ; END OF DTE ADDRESS
                                        ;
 000140  004567  000000G                DTEAEN: JSR     R5,$SAVRG               ; SAVE R3-R5
 000144  012700  000010                         MOV     #<DTEALN+1>/2,R0        ; GET LENGTH OF PACKED ADDRESS
 000150  012704  000000'                        MOV     #DTEADD,R4              ; GET ADDRESS OF DTE ADDRESS BUFFER
 000154  012705  000017'                        MOV     #DTEPCK,R5              ; GET ADDRESS OF PACKED ADDRESS BUFFER
 000160                                         CALL    PCKBCD                  ; PACK ADDRESS IN BCD FORMAT

 000164  012700  000027'                        MOV     #DTNTNM,R0              ; Clear DTE name to spaces          ;[TD001]
 000170  012701  000006                         MOV     #DTNMMX,R1                                                  ;[TD001]
 000174                                         CALL    SPFILL                                                      ;[TD001]
 000200  032767  000000G 000000G                BIT     #PF.XDF,$PFLAG          ; Check for extended data format    ;[TD001]
 000206  001024                                 BNE     20$                     ; If extended, cannot default DTE name ;[TD001]
                                                                                                                    ;[TD001]
 000210  016700  000000G                        MOV     $PSIPT,R0               ; POINT TO HOME BLOCK
 000214  062700  000024                         ADD     #H$NETW,R0              ; POINT TO DEFAULT NET NAME
 000220                                         GETRV   R0,#DTNMMX              ; READ IN NETWORK NODE NAME
 000236  012700  000000G                        MOV     #$BFR,R0                ; POINT TO NETWORK NODE NAME
 000242  012701  000027'                        MOV     #DTNTNM,R1              ; POINT TO LOCAL STORAGE OF NAME
 000246  012702  000006                         MOV     #DTNMMX,R2              ; GET SIZE OF NAME
 000252  112021                         10$:    MOVB    (R0)+,(R1)+             ; SET UP DEFAULT FOR DTE
 000254  005302                                 DEC     R2                      ; MORE TO SET UP?
 000256  003375                                 BGT     10$                     ; YES, SET AWAY
                                                                                                                    ;[TD001]
 000260                                 20$:    RETURN                          ; ELSE COMPLETED                    ;[TD001]
```

```
                                                                 D 13
VFGDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:28      PAGE 3
SYMBOL CROSS REFERENCE                                    CREF   04.00
SYMBOL   VALUE               REFERENCES

SF.LPB   = 000004            #5-67
SF.MFL   = 000040            #5-67
SF.PAC   = 000020            #5-67
SF.REA   = 000010            #5-67
SF.SER   = 000001            #5-67
SF.SVC   = 000002            #5-67
SF.UNL   = 000040            #5-67
SPACE    = 000040            #6-84
SPFILL   = ******   GX        12-282
STRNXT     000706 R           12-256      12-264     #19-470
SYNERR   = ******   GX       *9-157      *9-160
SYSFDB   = ******   GX       *12-288     *16-380     *16-389
S$$BAS   = ******             8-128       8-128       8-129       8-129       8-130       8-130       8-131       8-131       8-132
                              8-132
S.COST     000001            #5-67
S.FLG      000000            #5-67
S.LEN      000004            #5-67
S.NMST     000002            #5-67
S.OWNR     000003            #5-67
$ALPHA   = 000022            #10-173
$ANY     = 000020            #10-173
$BFR     = ******   GX        12-289      16-383      16-391
$BLANK   = 000006            #10-173
$CAT5    = ******   GX        18-432
$CDTB    = ******   GX        19-475
$CEACC   = ******   GX        16-388
$DIGIT   = 000024            #10-173
$DNUMB   = 000014            #10-173
$EOS     = 000012            #10-173
$ERR00     000046 R          #8-128       14-347
$ERR03     000106 R          #8-129       12-269      15-359
$ERR04     000140 R          #8-130       13-316
$ERR05     000170 R          #8-131       13-327
$ERR06     000224 R          #8-132       18-452
$ERR1T   = ******   GX        9-167
$EXIT    = 000000            #10-173
$FAIL    = 177777            #10-173
$GETAD   = ******   GX        16-389
$GETRV   = ******   GX        12-288      16-380
$GPRM    = ******             10-173
$LAMDA   = 000000            #10-173
$NUMBR   = 000002            #10-173
$PFLAG   = ******   GX        12-283
$PSIPT   = ******   GX        12-286      16-378
$QDTE      000000 RG         #9-151
$RADDR   = ******   GX       *12-288     *16-380     *16-389
$RAD50   = 000016            #10-173
$RBLCK   = ******   GX        16-381     *16-399
$RONLY   = ******             10-173      10-173      10-173
$SAVAL   = ******   GX        16-377
$SAVRG   = ******   GX        12-274
$STRNG   = 000004            #10-173
```

```
                                                  .SBTTL  PVC$DF ACTION ROUTINES


                                          ;
                                          ; PVC CIRCUIT ID
                                          ;
000056                                    PVNAM:                                                                        ;[TD001]
000056  012700  000036'                           MOV     #PVCNML,R0      ; ADDRESS OF BUFFER TO CONTAIN PVC NAME/COUNT  ;[TD001]
000062  012701  000006                            MOV     #PVNAMX,R1      ; MAX LENGTH FOR OLD-FORMAT PVC NAME           ;[TD001]
000066                                            CALL    STORID          ; STORE PVC NAME                               ;[TD001]
000072  103413                                    BCS     101$            ; BRANCH IF NAME TOO LONG                      ;[TD001]
000074  016767  000000G 000014'                   MOV     $X3DBS,PVCPSZ   ; SET UP DEFAULT MAXIMUM BLOCK SIZE            ;**-13
000102  016767  000000G 000016'                   MOV     $X3DWS,PVCWSZ   ; SET UP DEFAULT WINDOW SIZE
000110  005067  000020'                           CLR     PVCOWN          ; INITIALIZE PROCESS OWNER
000114  105067  000035'                           CLRB    PVCFLG          ; INITIALIZE FLAGS BYTE
000120                                            RETURN
                                          ;
                                          ; ERRORS
                                          ;
000122                                    101$:   EMSG$R  NO              ; ILLEGAL CIRCUIT ID

                                          ;
                                          ; LOGICAL CHANNEL NUMBER
                                          ;
000130  005767  000000G                   PVLCN:  TST     .PNUMH          ; LEGAL VALUE?
000134  001011                                    BNE     101$            ; BR IF NO
000136  016700  000000G                           MOV     .PNUMB,R0       ; GET SPECIFIED LOGICAL CHANNEL NUMER
000142  003406                                    BLE     101$            ; BR IF ILLEGAL VALUE
000144  020027  000000G                           CMP     R0,#CHNLMX      ; LEGAL VALUE?
000150  101003                                    BHI     101$            ; BR IF NO
000152  010067  000010'                           MOV     R0,PVCLCN       ; STORE LOGICAL CHANNEL NUMBER
000156                                            RETURN
                                          ;
                                          ; ERRORS
                                          ;
000160                                    101$:   EMSG$R  09              ; ILLEGAL LOGICAL CHANNEL NUMBER

                                          ;
                                          ; COUNTER TIMER
                                          ;
000166  005767  000000G                   PVCT:   TST     .PNUMH          ; LEGAL VALUE?
000172  001004                                    BNE     101$            ; BR IF NO
000174  016767  000000G 000012'                   MOV     .PNUMB,PVCCTM   ; SAVE COUNTER TIMER VALUE
000202                                            RETURN
                                          ;
                                          ; ERRORS
                                          ;
000204                                    101$:   EMSG$R  05              ; ILLEGAL COUNTER TIMER VALUE
```

```
Symbol table


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        002154    001   (RW,I,LCL,REL,CON)
DATA    000442    002   (RW,D,LCL,REL,CON)
$STATE  000212    003   (RW,D,LCL,REL,CON)
$KTAB   000006    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 87
Work  file writes: 86
Size of work file: 23533 Words  ( 92 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:01:08.16
SY:VFGPVC.V2,[132,134]VFGPVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFGPVC
```

```
    137                                              .SBTTL  ERROR MESSAGES
    138
    139                                              .ENABL  LC
    140 000056                                       NTLER$  ,3D,8B,CFERR,RTSPC,,<packet size>
    141 000102                                       NTLER$  ,3E,8B,CFERR,RTSPC,,<window size>
    142 000126                                       NTLER$  ,3F,8B,CFERR,RTSPC,,<flags word>
    143 000152                                       NTLER$  ,3G,8B,CFERR,RTSPC,,<maximum recalls>
    144 000202                                       NTLER$  ,3H,8B,CFERR,RTSPC,,<recall timer>
    145 000230                                       NTLER$  ,3I,8B,CFERR,RTSPC,,<DTE address>
    146 000254                                       NTLER$  ,3J,8B,CFERR,RTSPC,,<owner>
    147 000272                                       NTLER$  ,3K,8,CFERR,RTSPC,,<SVC descriptor allocation failure>
    148 000344                                       NTLER$  ,3M,8,CFERR,RTSPC,,<DTE address missing>
    149 000400                                       NTLER$  ,3Z,8,CFERR,RTSPC,,<PSI not loaded>
    150                                              .DSABL  LC
    151
    152 000000                                       .PSECT
```

```
                                                .SBTTL  FNDFIL - FIND FILTER BLOCK

                                        ;+
                                        ;
                                        ; FNDFIL - FIND FILTER BLOCK
                                        ;
                                        ; INPUTS:
                                        ;       CLASS - EVENT CLASS (IN BITS 6-15.)
                                        ;       RQB   - REQUEST BLOCK FOR SET/CLEAR LOGGING COMMAND
                                        ;
                                        ; OUTPUTS:
                                        ;       CARRY CLEAR:
                                        ;          $BFR - CONTENTS OF FILTER BLOCK
                                        ;          R0 - UNMAPPED ADDRESS OF MATCHING BLOCK
                                        ;          R1 - UNMAPPED ADDRESS OF PREVIOUS BLOCK
                                        ;       CARRY SET:
                                        ;          FILTER BLOCK NOT FOUND
                                        ;
                                        ;-
015534  012700  000000G                 FNDFIL: MOV     #$FILHD,R0              ; POINT TO FILTER BLOCK LISTHEAD
015540  011067  000000G                         MOV     (R0),$BFR               ; GET ADDRESS OF FIRST BLOCK IN LIST
015544  010067  000164'                         MOV     R0,CURUNM               ; INIT UNMAPPED ADDR OF CURRENT BLOCK
015550  016767  000164' 000166' 10$:            MOV     CURUNM,PRVUNM           ; SAVE UNMAPPED ADDR OF PREVIOUS
015556  000261                                  SEC                             ; ASSUME FAILURE
015560  016767  000000G 000164'                 MOV     $BFR,CURUNM             ; GET NEXT BLOCK IN LIST
015566  001474                                  BEQ     60$                     ; BR IF END OF LIST
015570                                          CEACC$  CURUNM,R0               ; CONVERT TO MAPPED ADDRESS
015602                                  20$:    GETAD   R0,#F.LEN               ; READ IN FILTER BLOCK
015620  016767  000002G 000000G                 MOV     F.CLS+$BFR,$RSIZE       ; GET REAL LENGTH OF BLOCK
015626  042767  177700  000000G                 BIC     #^C<FF.MSK>,$RSIZE      ; ...
015634  022767  000016  000000G                 CMP     #F.LEN,$RSIZE           ; IS THIS THE CORRECT LENGTH?
015642  001402                                  BEQ     25$                     ; BR IF YES
015644                                          GETAD                           ; ELSE READ IN ENTIRE FILTER BLOCK
015650  016746  000002G         25$:            MOV     F.CLS+$BFR,-(SP)        ; GET EVENT CLASS AND BLOCK SIZE
015654  042716  000077                          BIC     #FF.MSK,(SP)            ; ISOLATE EVENT CLASS
015660  026726  000074'                         CMP     CLASS,(SP)+             ; DOES THIS EVENT CLASS MATCH?
015664  001331                                  BNE     10$                     ; BR IF NO
015666  026767  000014G 000120'                 CMP     F.FLG+$BFR,EVQUAL       ; MATCH ON SINK TYPE OR QUALIFIERS?
015674  001325                                  BNE     10$                     ; BR IF NO
015676  032767  100000  000120'                 BIT     #FF.REM,EVQUAL          ; QUALIFIED BY SINK NODE?
015704  001404                                  BEQ     30$                     ; BR IF NO
015706  026767  000026G 000130'                 CMP     F.REM+$BFR,SINKND       ; MATCH ON SINK NODE ?
015714  001315                                  BNE     10$                     ; BR IF NO
015716  032767  000020  000120' 30$:            BIT     #FF.ADD,EVQUAL          ; QUALIFIED BY REMOTE NODE?
015724  001404                                  BEQ     40$                     ; BR IF NO
015726  026767  000016G 000132'                 CMP     F.ADD+$BFR,REMOTE       ; MATCH ON REMOTE NODE?
015734  001305                                  BNE     10$                     ; BR IF NO
015736  032767  000010  000120' 40$:            BIT     #FF.LIN,EVQUAL          ; QUALIFIED BY LINE?
015744  001404                                  BEQ     50$                     ; BR IF NO -- FOUND IT
015746  026767  000020G 000000G                 CMP     F.LIN+$BFR,$SLN         ; MATCH ON LINE?
015754  001275                                  BNE     10$                     ; BR IF NO -- KEEP LOOKING
015756  000241                          50$:    CLC
015760  016701  000166'                 60$:    MOV     PRVUNM,R1               ; GET UNMAPPED ADDRESS OF PREVIOUS
015764  016700  000164'                         MOV     CURUNM,R0               ; GET UNMAPPED ADDRESS OF CURRENT
015770                                          RETURN
```

```
                                  .SBTTL  INIEVT - INITIALIZE THE EVENT FILTER MASKS

                          ;+
                          ;
                          ; INIEVT - INITIALIZE THE EVENT MASKS
                          ;
                          ; INPUTS:
                          ;       R1 - ADDRESS OF EVENT FILTER MASKS IN FILTER CONTROL BLOCK
                          ;
                          ; OUTPUTS:
                          ;       R1 - POINTER TO END OF FILTER MASKS IN CONTROL BLOCK
                          ;
                          ;-

017454  005021            INIEVT: CLR     (R1)+                   ; CLEAR THE EVENT MASK FIELD IN BLOCK
017456  005021                    CLR     (R1)+                   ; ...
017460  005021                    CLR     (R1)+                   ; ...
017462  005021                    CLR     (R1)+                   ; ...
017464                            RETURN
```

```
                                                .SBTTL  RUN - RUN A TASK
                                        ;+
                                        ; **-RUN-RUN TASK
                                        ;
                                        ; INPUTS:
                                        ;       R4      - NUMBER OF TICKS TO DELAY FROM SYSTEM BOOT
                                        ;       R5      - TCB ADDRESS ( IN SYSTEM IMAGE )
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT SET IS NO NODE AVAILABLE
                                        ;       CLOCK QUEUE ENTRY GENERATED
                                        ;
                                        ;
                                        ; NOTE: THIS ALGORITHM WILL RUN THE TASK BEFORE ANY OTHER TASK
                                        ;               SCHEDULEO TO RUN WITH THE SAME DELAY FACTOR,
                                        ;               ( UNLIKE VMR RUN COMMAND ).
                                        ;
                                        ;-

021232  012701  000020                  RUN:    MOV     #C.LGTH,R1      ; SIZE OF CLOCK QUEUE ENTRY
021236                                          CALL    $ALL1           ; ALLOCATE SPACE FROM RSX11M POOL
021242  103513                                  BCS     100$            ; NO SPACE AVAILABLE

021244  010046                                  MOV     R0,-(SP)        ; SAVE NODE ADDRESS
021246  010546                                  MOV     R5,-(SP)        ; AND TCB ADDRESS
021250  010446                                  MOV     R4,-(SP)        ; AND # OF TICKS

021252  012704  000000G                         MOV     #$BFR,R4        ; R4 POINTS AT VIRTUAL I/O BUFFER
021256  012705  000050'                         MOV     #CLKQB,R5       ; R5 AT CLOCK QUEUE ENTRY

021262  012665  000006                          MOV     (SP)+,C.TIM(R5) ; SET NUMBER OF TICKS
021266  012765  000004  000002                  MOV     #C.SSHT,C.RQT(R5) ; SINGLE SHOT
021274  005065  000012                          CLR     C.RSI(R5)       ; NO RESCHEDULE INT. COUNT
021300  005065  000014                          CLR     C.RSI+2(R5)     ; NO RSI UNITS
021304  012665  000004                          MOV     (SP)+,C.TCB(R5) ; SET TCB ADDRESS
021310                                          GETRV   .ABTIM,#2       ; GET ABSOLUTE TIME COUNTER
021330  061465  000006                          ADD     (R4),C.TIM(R5)  ; FORM ABSOLUTE TIME OF REQUEST
021334  005565  000010                          ADC     C.TIM+2(R5)     ; ...

                                        ; INSERT NEWLY FORMED ENTRY INTO CLOCK QUEUE IN TEMPORAL SEQUENCE

021340  0[illegible]702  000000G                         MOV     .CLKHD,R2       ; PREVIOUS ENTRY
021344                                          GETRV   R2,#C.LGTH      ; READ IN LISTHEAD

021362  010203                          10$:    MOV     R2,R3           ; SAVE PREVIOUS ENTRY ADDRESS
021364  011402                                  MOV     (R4),R2         ; GET ADDRESS OF NEXT ENTRY
021366  001421                                  BEQ     30$             ; END OF LIST
021370                                          GETRV   R2              ; READ IN NEXT CLOCK QUEUE ENTRY
021400  026564  000004  000004                  CMP     C.TCB(R5),C.TCB(R4) ; TASK ALREADY ACTIVE ?
021406  001432                                  BEQ     101$            ; YES
021410  026564  000010  000010                  CMP     C.TIM+2(R5),C.TIM+2(R4) ; NEW ENTRY AT LATER TIME ?
021416  101361                                  BHI     10$             ; NO
021420  103404                                  BLO     30$             ; YES
021422  026564  000006  000006                  CMP     C.TIM(R5),C.TIM(R4) ; MAYBE
021430  101354                                  BHI     10$             ; PREMPT IF TIME IS SAME

021432  010215                          30$:    MOV     R2,(R5)         ; POINT OUR NEW ENTRY AT NEXT
021434  012600                                  MOV     (SP)+,R0        ; GET OUR ENTRY'S ADDRESS
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 3
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL   VALUE            REFERENCES

D$MLN      000030          *74-4011    74-4026
D$OUTT     000043          *26-1325
D$RETF     000050          *26-1359
D$RNN      000002           27-1402   *27-1404    28-1577    28-1579
D$SEG      000036          *26-1291
D.NAM    = ******   GX      77-4165
D.PCB    = ******   GX      77-4164
D.UCB    = ******   GX      38-1911    49-2773    50-2830
EF$ACT   = 000001          #5-79
ENALIN     003134 R         11-489    #23-1064
ERR1       000336 R        #9-218      11-514
ERR10      001040 R        #9-238      39-2058
ERR12      001104 R        #9-240      39-2063    40-2177
ERR13      001136 R        #9-242      39-2069    82-4356
ERR14      001176 R        #9-244      39-2064    40-2178
ERR15      001244 R        #9-246      39-2062    40-2176
ERR16      001314 R        #9-248      39-2065    40-2179
ERR17      001342 R        #9-250      27-1465    28-1591
ERR18      001414 R        #9-252      27-1466    28-1592    41-2284
ERR19      001442 R        #9-254      27-1467    28-1593    42-2438
ERR2       000372 R        #9-220      11-515
ERR20      001472 R        #9-256      41-2283
ERR21      001520 R        #9-258      11-522     16-754     17-803     20-941     21-996     22-1049    24-1154
ERR22      001572 R        #9-260      16-756
ERR23      001636 R        #9-262      38-1926
ERR24      001664 R        #9-264      42-2436
ERR25      001726 R        #9-266      42-2437
ERR26      001762 R        #9-268      42-2313    42-2363    43-2469
ERR27      002020 R        #9-270      44-2534
ERR28      002054 R        #9-272      46-2664
ERR29      002110 R        #9-274      28-1594
ERR3       000424 R        #9-222      38-1925    83-4393
ERR30      002144 R        #9-276      41-2285
ERR31      002176 R        #9-278      13-614     14-646     15-678
ERR32      002236 R        #9-280      13-615
ERR33      002306 R        #9-282      14-647
ERR34      002354 R        #9-284      15-679
ERR35      002402 R        #9-286      16-753     17-802     20-940     21-995     22-1048
ERR36      002444 R        #9-288      16-752
ERR37      002476 R        #9-290      16-755
ERR38      002546 R        #9-292      57-3228    59-3353
ERR39      002614 R        #9-294      47-2712
ERR4       000462 R        #9-224      83-4396
ERR40      002670 R        #9-296      54-3073
ERR41      002742 R        #9-298      53-3042
ERR42      003000 R        #9-300      47-2713
ERR43      003050 R        #9-302      80-4230
ERR44      003122 R        #9-304      79-4210
ERR45      003152 R        #9-306      40-2170
ERR46      003226 R        #9-308      40-2172
ERR47      003300 R        #9-310
ERR48      003350 R        #9-312
```

```
VENA      CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 16
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE              REFERENCES

T$LLEN     000046           #5-68       81-4275     81-4280
T$LOPR     000002           #5-68
T$LTCL     000024           #5-68
T$LTIM     000026           #5-68
T$LTPR     000014           #5-68
T$LTPS     000020           #5-68
T$NAPL     000004           #5-68
T$NFE      000000           #5-68
T$NLEN     000010           #5-68
T$NNUL     000002           #5-68
T$NOPL     000006           #5-68
T$NRNI     000042           #5-68
T$NRPL     000005           #5-68
T$NRUL     000007           #5-68
T$NVR      000001           #5-68
T$RPRI     000040           #5-68
T$SVC      000034           #5-68
T$T5       000030           #5-68      *80-4227
T$T6       000032           #5-68      *79-4207
T.ACT      000031           #5-82
T.CHA      000032           #5-82
T.CSR      000022           #5-82
T.ITM      000001           #5-82
T.NAM      000006            84-4423    84-4425
T.QUE      000024           #5-82        5-82
T.QU2    = 000026           #5-82
T.RCVL     000012            39-1991    39-1993    *39-2001    *39-2002     40-2134     40-2136    *40-2143    *40-2144     75-4092
                             76-4140    76-4142
T.RTY      000030           #5-82
T.ST2      000034            39-1959    39-1971     40-2102     40-2118
T.TCBL     000030           *84-4417    84-4419
T.TMR      000000           #5-82
T.TRLG   = 000076            22-1037
T.TRLN     000005            22-1038
T.TRPN     000004           *13-607
T2.FXD   = 002000            39-1959    39-1971     40-2102     40-2118
U.CNT      000030            49-2774    50-2831
U.CW1      000010            49-2779    50-2836
U.CW3      000014            38-1912    38-1913    *38-1915    *38-1918
U.RED      000002            49-2775    49-2777     50-2832     50-2834
VALI       011734 R          42-2309   #43-2464
VF$RCV   = 100000            36-1840    36-1850
VF$XMT   = 040000            37-1878    37-1888
VF.ACP   = ******   GX       39-2024
VF.ADD   = ******   GX       26-1264    28-1490     28-1504     42-2365
VF.DLX   = ******   GX       11-414     11-479      11-494      11-510      23-1065     24-1098     39-1940     39-1984     83-4387
VF.EAD   = ******   GX       54-3061
VF.FIX   = ******   GX       39-2010    40-2152
VF.ON    = ******   GX       11-474     11-518      39-1984     40-2129     55-3100
VF.PEM   = ******   GX       39-1940    83-4387     83-4394
VF.SED   = ******   GX       12-539
VF.SKA   = ******   GX       53-3025
```

```
VEXP      CREATED BY  MACRO  ON 29-JUN-85 AT 02:27     PAGE 1
SYMBOL CROSS REFERENCE                                 CREF   04.00

SYMBOL   VALUE              REFERENCES

FL.CHA   = ******  GX         5-76
$ADDR    = ******  GX         5-72
$FLAGS   = ******  GX         5-76
$FLAG1   = ******  GX         5-74
$NALL    = ******  GX         5-81
$VEXP      000000 RG         #5-71
$VEX2    = ******  GX         5-78
$VEX3    = ******  GX         5-83
$VEX5    = ******  GX         5-86
```

```
                                   ;
                                   ; POINTER TO ALLOCATED CORE
                                   ;
000774  112524                     CORE:   MOVB    (R5)+,(R4)+         ; COPY FIRST WORD OF ADDRESS
000776  112524                             MOVB    (R5)+,(R4)+         ;       ...

001000  005767  000000G                    TST     .MGMGE              ; MEMORY MANAGEMENT SYSTEM ?
001004  100402                             BMI     5$                  ; NO

001006  112524                             MOVB    (R5)+,(R4)+         ; COPY SECOND WORD OF ADDRESS
001010  112524                             MOVB    (R5)+,(R4)+         ;       ...

001012  112546                     5$:     MOVB    (R5)+,-(SP)         ; ASSEMBLE COUNT
001014  112566  000001                     MOVB    (R5)+,1(SP)         ;       ...
001020  012603                             MOV     (SP)+,R3            ;       ...
001022  001417                             BEQ     30$                 ; IF ZERO, NO UNLOAD DATA TO STORE

001024  016702  000000G                    MOV     $UDA,R2             ; GET UNLOAD DATA ADDRESS

001030  005767  000000G                    TST     .MGMGE              ; MEMORY MANAGEMENT ?
001034  100403                             BMI     10$                 ; NO

001036  016422  177774                     MOV     -4(R4),(R2)+        ; STORE ALLOCATION BIAS
001042  000402                             BR      20$

001044  016422  177776             10$:    MOV     -2(R4),(R2)+        ; STORE ALLOCATION ADDRESS
001050  006303                     20$:    ASL     R3                  ; SHIFT COUNT LEFT 1 PLACE
001052  005203                             INC     R3                  ; MAKE IT ODD
001054  010322                             MOV     R3,(R2)+            ; STORE COUNT
001056  010267  000000G                    MOV     R2,$UDA             ; SET NEW UNLOAD DATA [illegible]DRESS
001062                             30$:    RETURN
```

```
VEX2      CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE            REFERENCES

ADDRB     000062 R         5-93      #7-200
ADDRW     000102 R         5-94      #7-208
APR       002212 R         5-124     #13-772
BLKB      000130 R         5-95      #7-219
BLKW      000142 R         5-96      #7-228
BYTE      000154 R         5-97      #7-237
CORE      000774 R         5-98      #8-444
CSR       000160 R         5-99      #7-243
CTIM      000172 R         5-100     #7-251
DPRB      002224 R         5-125     #13-779
DPRW      002232 R         5-126     #13-782
DVCHA     000200 R         5-101     #7-257
D$END   = 000104           7-306      7-373
D$LNUM    000014           7-375
D$MLL     000040           7-307
EOF       000356 R         5-102     #7-300
FILE      001212 R         5-103     #11-520
FLAGA     000004 R        #5-82      *12-599    *12-604     12-614    *12-616     12-665
FLAGS     000450 R        #5-154     *6-177      7-200      7-208      7-262      7-289      7-330      7-348      7-359
                           7-389      7-402      7-417      7-427     *11-519    *11-536    *11-542    *11-570    *13-713
                          *13-758
FL.DDM  = ******   GX      7-258      7-285
FL.KMX  = ******   GX      7-333
FL.MUX  = ******   GX      7-260      7-287
F.NRBD  = ******   GX     *12-667    *13-747    *13-755
F.PRLT  = 000001          #5-83       12-604     12-614
F.RSIZ  = ******   GX     *7-306     *7-373     *12-670    *13-748
F.UFBD  = ******   GX     *7-305     *7-309
F.WRIT  = 000002          #5-84       12-616     12-665
ICBL      000212 R        #5-87      *12-605     12-617    *12-657
ICBLK     000006 R        #5-86       5-87       12-605     12-667
ICBPT     000002 R        #5-79      *12-606     12-658    *12-659
INDEX     000446 R        #5-149     *6-176      7-264      7-291      7-332      7-350      7-361      7-393      7-406
                           7-419      7-429     *11-531     11-532    *11-535    *11-565     11-566    *11-569    *13-732
                           13-733    *13-757     13-772
INT       001122 R         5-104     #10-493
JSROF   = 000012          #5-80       12-654
KRBB      000214 R        #5-88
LB.ADR  = ******   GX      12-585
LFILE     001304 R         5-107     #11-543
LF.ACT  = 100000          #5-60
LF.BRO  = 000400          #5-60
LF.BWT  = 000007          #5-60
LF.ENA  = 002000          #5-60
LF.LPB  = 001000          #5-60
LF.MDC  = 000100          #5-60
LF.MFL  = 004000          #5-60
LF.MTP  = 000020          #5-60
LF.PAC  = 000200          #5-60
LF.RDY  = 040000          #5-60
LF.REA  = 010000          #5-60
LF.SER  = 000040          #5-60
```

```
                                         ;
                                         ; LIBRARY POINTER
                                         ;
 000146                                  XLIBR:
 000146  112546                                  MOVB    (R5)+,-(SP)     ; ASSEMBLE BIAS
 000150  112566  000001                          MOVB    (R5)+,1(SP)     ;       ...
 000154  012600                                  MOV     (SP)+,R0        ;       ...
 000156  112546                                  MOVB    (R5)+,-(SP)     ; AND OFFSET
 000160  112566  000001                          MOVB    (R5)+,1(SP)     ;       ...
 000164  012602                                  MOV     (SP)+,R2        ;       ...
 000166  005767  000000G                         TST     .MGMGE          ; MAPPED TARGET SYSTEM?
 000172  100401                                  BMI     10$             ; IF MI, NO
 000174  122525                                  CMPB    (R5)+,(R5)+     ; DISCARD EXTRA WORD
 000176                                  10$:    CALLR   $XLBR           ; RELEASE LIBRARY REFERENCE
```

```
000430  103410                          BLO     5$              ; NO IF LO
000432  010346                          MOV     R3,-(SP)        ; LINE TABLE ADDRESS
000434  042716  160000                  BIC     #160000,(SP)    ; CLEAR THE APR BITS
000440  012677  000000G                 MOV     (SP)+,@$ADRZ    ; NEW VIRTUAL ADDRESS
000444  016777  000000G 000000G         MOV     $BIAS,@$ADRX    ; BIAS OF PROCESS
000452                          5$:     GETAD   ,#2             ; ACCESS LINE TABLE
000464  016767  000000G 000000G         MOV     $$PCHA,$BFR     ; COPY PROTOCOL EMULATOR CHARACTERISTICS
000472                                  PUTAD                   ; RE-WRITE LINE TABLE
000476  062703  000002          10$:    ADD     #2,R3
000502                                  RETURN

                                ;
                                ; END OF FILE
                                ;
000504                          EOF:    CRASH


                                ;
                                ; INTERRUPT ENTRY POINT
                                ;
000506  122525                  INT:    CMPB    (R5)+,(R5)+
000510  005767  000000G                 TST     .MGMGE          ; MAPPED TARGET SYSTEM ?
000514  100403                          BMI     10$             ; NO ..
000516  062703  000016                  ADD     #14.,R3
000522  000402                          BR      20$
000524  062703  000004          10$:    ADD     #4,R3
000530                          20$:    RETURN


                                ;
                                ; LIBRARY POINTER
                                ;
000532  005767  000000G         LIBR:   TST     .MGMGE          ; MAPPED TARGET SYSTEM ?
000536  100405                          BMI     10$             ; NO ..
000540  062705  000006                  ADD     #6,R5
000544  062703  000004                  ADD     #4,R3
000550  000404                          BR      20$
000552  062705  000004          10$:    ADD     #4,R5
000556  062703  000002                  ADD     #2,R3
000562                          20$:    RETURN


                                ;
                                ; LOGICAL LINKS
                                ;
000564  062703  000002          LINKS:  ADD     #2,R3
000570                                  RETURN


                                ;
                                ; LISTHEAD
                                ;
000572  062703  000004          LSTHD:  ADD     #4,R3
000576                                  RETURN


                                ;
                                ; LINE TABLE ADDRESS
                                ;
000600  062703  000002          LTAB:   ADD     #2,R3
000604                                  RETURN

```

```
VEX5      CREATED BY  MACRO  ON 29-JUN-85 AT 02:28     PAGE 3
SYMBOL CROSS REFERENCE                                 CREF   04.00

SYMBOL   VALUE            REFERENCES

TABLE      000000 R      #5-83      6-191      8-490      8-518      9-560
TIME       001444 R       5-117    #9-578
UBA        000160 R      #5-151      6-181    *10-624    *10-669
UBIAS      000156 R      #5-150    *10-670
UNB        001022 R       5-111    #7-456
UNW        001026 R       5-112    #7-462
WORD       001034 R       5-113    #7-468
X2CH01     001750 R       5-131    #12-738
X2CH02     001754 R       5-132    #12-743
X2CH03     001760 R       5-133    #12-748
X2CH04     001764 R       5-134    #12-753
X2CH05     001770 R       5-135    #12-758
X3CH01     001674 R       5-120    #11-681
X3CH02     001700 R       5-121    #11-686
X3CH03     001704 R       5-122    #11-691
X3CH04     001710 R       5-123    #11-696
X3CH05     001714 R       5-124    #11-701
X3CH06     001720 R       5-125    #11-706
X3CH07     001724 R       5-126    #11-711
X3CH08     001730 R       5-127    #11-716
X3CH09     001734 R       5-128    #11-721
X3CH10     001740 R       5-129    #11-726
X3CH11     001744 R       5-130    #11-731
$ADDR    = ******   GX    6-177
$ADRX    = ******   GX    7-281      7-289      7-310      7-318      7-414      7-422      9-533      9-541      10-629
                          10-663     10-670
$ADRZ    = ******   GX    7-280      7-288      7-309      7-317      7-413      7-421      9-532      9-540      10-628
$BFR     = ******   GX   *7-296     *7-297     *7-320     *7-424      9-543      10-673
$BIAS    = ******   GX    7-289      7-318      7-422      9-541
$BIN     = ******   GX    6-176
$FLAGS   = ******   GX    7-407
$GETAD   = ******   GX    7-290      7-319      7-423      9-542      10-632
$LINK    = ******   GX    7-294
$NTLHB   = ******   GX    10-629
$PTR     = ******   GX    6-185
$PUTAD   = ******   GX    7-298      7-321      7-425
$RSIZE   = ******   GX   *10-626
$SLTA    = ******   GX    6-178      6-183      7-292
$VEX5      000000 RG     #6-173
$$DCHA   = ******   GX    7-296      7-297
$$PCHA   = ******   GX    7-320
$$SCSR   = ******   GX    7-424
.CXUNL   = ******   GX    10-629
.EOF     = ******   GX    8-487      8-515      9-557
.MGMGE   = ******   GX    7-251      7-282      7-311      7-334      7-344      7-415      9-534
```

```
                           ;
                           ; HASH TABLE SIZE
                           ;
000262  005767  000000G    HSHTBL: TST     .PNUMH          ; LEGAL HASH TABLE SIZE VALUE?
000266  001016                     BNE     101$            ; BR IF NO
000270  016700  000000G            MOV     .PNUMB,R0       ; GET HASH TABLE SIZE
000274  012701  000040             MOV     #HSHMN,R1       ; GET MINIMUM SIZE OF HASH TABLE
000300  020001             10$:    CMP     R0,R1           ; IS THIS A LEGAL SIZE?
000302  001405                     BEQ     20$             ; BR IF YES
000304  006301                     ASL     R1              ; HASH TABLE SIZE MUST BE POWER OF TWO
000306  020127  001000             CMP     R1,#HSHMX       ; IS THIS A LEGAL SIZE?
000312  101004                     BHI     101$            ; BR IF NO
000314  000771                     BR      10$             ; CHECK FOR LEGAL VALUE
000316  010067  000000G    20$:    MOV     R0,HSHSZ        ; SAVE HASH TABLE SIZE
000322                             RETURN
                           ;
                           ; ERRORS
                           ;
000324                     101$:   EMSG$R  04              ; ILLEGAL HASH TABLE SIZE

                           ;
                           ; COUNTER TIMER VALUE
                           ;
000332  005767  000000G    CNTTIM: TST     .PNUMH          ; LEGAL COUNTER TIMER VALUE?
000336  001001                     BNE     101$            ; BR IF NO
000340                             RETURN
                           ;
                           ; ERRORS
                           ;
000342                     101$:   EMSG$R  05              ; ILLEGAL COUNTER TIMER VALUE

```

```
VFGDTE     CREATED BY  MACRO  ON 29-JUN-85 AT 02:28     PAGE 4

SYMBOL CROSS REFERENCE                                 CREF   04.00

SYMBOL  VALUE              REFERENCES

$SUBXP  = 000010           #10-173
$$$FLG  = 177777           #10-173
$$$KEY  = 177777           #10-173
.PCHAR  = ******   GX        19-470       19-472
.PNUMB  = ******   GX        13-303       18-440       18-447
.PNUMH  = ******   GX        13-301       13-321       18-438       18-445
.PSTCN  = ******   GX        14-334       18-429
.PSTPT  = ******   GX        14-337       18-431
.TPARS  = ******   GX         9-158
```

```

                                         ;
                                         ; MAXIMUM BLOCK SIZE
                                         ;
000212                                   PVPSZ:  CALL    CHKBLK          ; CHECK FOR LEGAL BLOCK SIZE VALUE
000216  103403                                   BCS     101$            ; BR IF ERROR
000220  010067  000014'                          MOV     R0,PVCPSZ       ; SAVE MAXIMUM BLOCK SIZE
000224                                           RETURN
                                         ;
                                         ; ERRORS
                                         ;
000226                                   101$:   EMSG$R  P8              ; ILLEGAL MAXIMUM BLOCK SIZE

                                         ;
                                         ; WINDOW SIZE
                                         ;
000234                                   PVWSZ:  CALL    CHKWND          ; CHECK FOR LEGAL WINDOW SIZE VALUE
000240  103403                                   BCS     101$            ; BR IF ILLEGAL VALUE
000242  010067  000016'                          MOV     R0,PVCWSZ       ; SAVE WINDOW SIZE
000246                                           RETURN
                                         ;
                                         ; ERRORS
                                         ;
000250                                   101$:   EMSG$R  00              ; ILLEGAL WINDOW SIZE

                                         ;
                                         ; PROCESS OWNER
                                         ;
000256  022767  000003  000000G          PVOWN:  CMP     #3,.PSTCN       ; LEGAL PROCESS OWNER?
000264  103410                                   BLO     101$            ; BR IF NO
000266  016700  000000G                          MOV     .PSTPT,R0       ; POINT TO STRING TO CONVERT TO RAD50
000272  005001                                   CLR     R1              ; PERIOD IS NOT A VALID TERMINATOR
000274                                           CALL    $CAT5           ; CONVERT TO RAD50
000300  010067  000020'                          MOV     R0,PVCOWN       ; SAVE PROCESS OWNER NAME
000304                                           RETURN
                                         ;
                                         ; ERRORS
                                         ;
000306                                   101$:   EMSG$R  N1              ; ILLEGAL PROCESS OWNER NAME

                                         ;
                                         ; FLAGS BYTE
                                         ;
000314  005767  000000G                  PVFG:   TST     .PNUMH          ; LEGAL VALUE?
000320  001007                                   BNE     101$            ; BR IF NO
000322  105767  000001G                          TSTB    .PNUMB+1        ; LEGAL VALUE?
000326  001004                                   BNE     101$            ; BR IF NO
000330  116767  000000G 000035'                  MOVB    .PNUMB,PVCFLG   ; SAVE FLAGS BYTE
000336                                           RETURN
                                         ;
                                         ; ERRORS
                                         ;
000340                                   101$:   EMSG$R  N2              ; ILLEGAL FLAGS BYTE VALUE
```

```
VFGPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:30      PAGE 1
SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL  VALUE             REFERENCES

ALLXCB    000434 R          13-414    #14-438
BIASX     001466 R          16-547    #18-626
CERR    = ******  GX        8-165      8-166     8-167     8-168     8-169     8-170     8-171     8-172
CFGBF   = ******  GX        9-195
CFGSZ   = ******  GX        9-194
CFLIN   = ******  GX        8-165      8-166     8-167     8-168     8-169     8-170     8-171     8-172
CHKBLK  = ******  GX        12-350
CHKWND  = ******  GX        12-362
CHNLMX  = ******  GX        11-325
DLMCTL    000057 R         #7-150     *19-665
DLMIND    000061 R         #7-152     *19-654
DLMPDV    000022 R         #7-140     *19-655
DLMSLN    000030 R         #7-143      14-466   *20-708   *20-709
DLMSLT    000024 R         #7-141
DLMSTA    000026 R         #7-142      13-404   *19-677    20-713
DLMTLC    001662 R          13-413    #20-703
DLMUNT    000060 R         #7-151
DTEDES  = ******  GX        16-551     17-586
E$NBR     000014           #5-66
E$NBS     000020           #5-66
E$NCR     000034           #5-66
E$NCS     000036           #5-66
E$NIC     000044           #5-66
E$NLEN    000050           #5-66
E$NLLA    000012           #5-66
E$NLNK    000000           #5-66
E$NML     000040           #5-66
E$NMR     000024           #5-66
E$NMS     000030           #5-66
E$NNOD    000002           #5-66
E$NRT     000042           #5-66
E$NRTP    000005           #5-66
E$NSEG    000010           #5-66
E$NTIM    000046           #5-66
E$NUSE    000004           #5-66
E$STPT    000006           #5-66
FLAG      000032 R         #7-144     *9-190
FMT8    = ******  GX        8-168
FMT8B   = ******  GX        8-165      8-166     8-167     8-169     8-170     8-171     8-172
FM.8    = 000000           #8-168
FM.8B   = 000000           #8-165     #8-166    #8-167    #8-169    #8-170    #8-171    #8-172
FNDHSH    001304 R          14-457    #17-585
FNDPDV  = ******  GX        19-652
FNDPRT    000664 R          13-411    #15-490
F.RSIZ  = ******  GX       *14-446    *14-452   *14-463   *15-494   *15-497   *15-501   *16-541   *17-587   *17-598
                           *20-731    *20-736
HSHADD  = ******  GX       *17-588     17-597
HSHSZ   = ******  GX        17-591     17-602
HTIMR     000000 R         #7-131     *13-406    20-747
H$NPT     000022            15-493     15-496
H$PTB     000020            14-445     15-496
INIXCB    001062 R          14-456    #16-536
```

```
                                        .SBTTL  $QSVC - LOOK FOR SVC$DF MACRO

                                ;+
                                ;
                                ; $QSVC - LOOK FOR SVC$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ALL REGISTERS DESTROYED
                                ;
                                ;-
000000  012705  000000'         $QSVC:: MOV     #SVCDF,R5       ; STATE TABLE ADDRESS
000004  005001                          CLR     R1              ; FULL KEYWORD MATCH LENGTH
000006  012702  000000'                 MOV     #SVCKW,R2       ; KEYWORD TABLE ADDRESS
000012  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
000016  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000022  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000026                                  CALL    .TPARS          ; GO DO THE PARSE
000032  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000034  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000040  001401                          BEQ     101$            ; IF EQ, YES
000042                          20$:    RETURN
                                ;
                                ; ERRORS
                                ;
000044                          101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
                                          .SBTTL  FILOPR - OPERATE ON FILTER CONTROL BLOCK

                                  ;+
                                  ;
                                  ; FILOPR - OPERATE ON FILTER CONTROL BLOCK
                                  ;
                                  ; INPUTS:
                                  ;       R0 - ADDRESS OF MATCHING BLOCK IN LIST
                                  ;       R2 - ADDRESS OF GLOBAL EVENT FILTER BLOCK FOR THIS CLASS
                                  ;
                                  ; OUTPUTS:
                                  ;       THE EVENT FILTERS ARE SET ACCORDINGLY
                                  ;       CARRY CLEAR:
                                  ;          CURRENT FILTER BLOCK CONTAINS USEFUL INFORMATION
                                  ;       CARRY SET:
                                  ;          NO USEFUL INFORMATION IN FILTER BLOCK (ALL EVENT MASKS ARE CLEAR)
                                  ;
                                  ;-

015772 010046                     FILOPR: MOV     R0,-(SP)               ; SAVE R0
015774 032767 000000G 000000G             BIT     #CM$SET,$CMAND         ; IS THIS A SET COMMAND?
016002 001557                             BEQ     20$                    ; BR IF NO
                                  ;
                                  ; SET EVENT FILTER OPERATION
                                  ;
016004 012746 000016                      MOV     #F.LEN,-(SP)           ; ASSUME GLOBAL FILTER BLOCK
016010 020002                             CMP     R0,R2                  ; OPERATE ON GLOBAL FILTER
016012 001441                             BEQ     10$                    ; BR IF YES
016014 012716 000040                      MOV     #F.CEV+10,(SP)         ; GET LENGTH OF NON-GLOBAL FILTER BLOCK
016020 012703 000004G                     MOV     #RQB+LO.EVT,R3         ; POINT TO FILTER MASKS
016024 012705 000030G                     MOV     #$BFR+F.CEV,R5         ; POINT TO CURRENT FILTER BLOCK MASKS
016030                                    CALL    CLREVT                 ; CLEAR THE 'CLEAR' EVENT MASKS

016034 005702                             TST     R2                     ; WAS A GLOBAL FILTER BLOCK FOUND?
016036 001427                             BEQ     10$                    ; BR IF NO
016040 012703 000106'                     MOV     #GMASK,R3              ; POINT TO GLOBAL FILTER MASKS
016044 012705 000004G                     MOV     #RQB+LO.EVT,R5         ; POINT TO FILTER MASKS
016050                                    CALL    CLREVT                 ; CLEAR ANY MASKS SET IN GLOBAL BLOCK
016054 103020                             BCC     10$                    ; BR IF WORK TO DO
016056                                    SAVRG   <R0>                   ; SAVE UNMAPPED ADDRESS
016060                                    CEACC$  R0                     ; CONVERT TO MAPPED ADDRESS
016070                                    PUTAD   R0,2(SP)               ; WRITE UPDATED FILTER BLOCK
016106                                    RESRG   <R0>                   ; RESTORE UNMAPPED ADDRESS
016110 005726                             TST     (SP)+                  ; CLEAN UP STACK
016112 000167 000420                      JMP     40$                    ; BR IF NO WORK TO DO
                                  ;
                                  ; OPERATE ON GLOBAL FILTER BLOCK
                                  ;
016116 012705 000004G             10$:    MOV     #$BFR+F.SEV,R5         ; POINT TO CURRENT FILTER BLOCK MASKS
016122 012703 000004G                     MOV     #RQB+LO.EVT,R3         ; POINT TO FILTER MASKS
016126                                    CALL    SETEVT                 ; SET THE FILTER MASKS
016132                                    SAVRG   <R0>                   ; SAVE UNMAPPED ADDRESS
016134                                    CEACC$  R0                     ; CONVERT TO MAPPED ADDRESS
016144                                    PUTAD   R0,2(SP)               ; WRITE UPDATED FILTER BLOCK
016162                                    RESRG   <R0>                   ; RESTORE UNMAPPED ADDRESS
016164 005726                             TST     (SP)+                  ; CLEAN UP STACK
016166 020002                             CMP     R0,R2                  ; OPERATE ON GLOBAL FILTER BLOCK?
```

```
 3876                                                   .SBTTL  SETQLF - SET QUALIFIER FIELDS IN FILTER BLOCK
 3877
 3878                                           ;+
 3879                                           ;
 3880                                           ; SETQLF - SET QUALIFIER FIELDS IN FILTER BLOCK
 3881                                           ;
 3882                                           ; INPUTS:
 3883                                           ;       R1 - ADDRESS OF FIRST QUALIFIER FIELD IN FILTER BLOCK
 3884                                           ;
 3885                                           ; OUTPUTS:
 3886                                           ;       R1 - POINTS PAST END OF QUALIFIER FIELDS IN FILTER BLOCK
 3887                                           ;
 3888                                           ;-
 3889
 3890 017466  010146                            SETQLF: MOV     R1,-(SP)                        ; SAVE FILTER BLOCK ADDRESS
 3891 017470  162716  000002                            SUB     #2,(SP)                         ; POINT TO FLAGS WORD
 3892 017474  032767  000020  000000G                   BIT     #OP$NOD,$OPTON                  ; REMOTE NODE QUALIFIER?
 3893 017502  001403                                    BEQ     10$                             ; BR IF YES
 3894 017504  016721  000132'                           MOV     REMOTE,(R1)+                    ; STORE REMOTE NODE ADDRESS
 3895 017510  000401                                    BR      20$                             ; CONTINUE
 3896 017512  005021                            10$:    CLR     (R1)+                           ; NO REMOTE NODE QUALIFIER
 3897 017514  032767  000040  000000G           20$:    BIT     #OP$LNE,$OPTON                  ; LINE QUALIFIER?
 3898 017522  001405                                    BEQ     30$                             ; BR IF NO
 3899 017524  116721  000000G                           MOVB    $SLN,(R1)+                      ; STORE SLN
 3900 017530  116721  000034G                           MOVB    RQB+LO.TRB,(R1)+                ; STORE STATION NUMBER
 3901 017534  000401                                    BR      40$                             ; CONTINUE
 3902 017536  005021                            30$:    CLR     (R1)+                           ; NO LINE QUALIFIER
 3903 017540  005021                            40$:    CLR     (R1)+                           ; NO CIRCUIT QUALIFIER
 3904 017542  005021                                    CLR     (R1)+                           ; NO MODULE QUALIFIER
 3905 017544  032767  000010  000000G                   BIT     #OP$SIN,$OPTON                  ; SINK NODE QUALIFIER?
 3906 017552  001407                                    BEQ     50$                             ; BR IF NO
 3907 017554  016721  000130'                           MOV     SINKND,(R1)+                    ; STORE SINK NODE ADDRESS
 3908 017560  002005                                    BGE     60$                             ; IF NOT $HOST BRANCH
 3909 017562  052776  040000  000000                    BIS     #FF.HST,@(SP)                   ; SET HOST FLAG
 3910 017570  000401                                    BR      60$                             ; CONTINUE
 3911 017572  005021                            50$:    CLR     (R1)+                           ; NO SINK NODE QUALIFIER
 3912 017574  005726                            60$:    TST     (SP)+                           ; ADJUST STACK
 3913
 3914 017576                                            RETURN
```

```
   4344 021436                                     GETRV   R3              ; READ IN PREVIOUS ENTRY
   4345 021446  010014                             MOV     R0,(R4)         ; POINT PREVIOUS ENTRY AT NEW ONE
   4346 021450                                     PUTRC                   ; REWRITE PREVIOUS ENTRY
   4347 021454                                     SWTCHO  #CLKQB          ; POINT AT BUFFERED NEW ENTRY
   4348 021462                                     PUTRC   R0              ; WRITE IT TO APPROPRIATE ADDRESS
   4349 021472                             100$:   RETURN
   4350
   4351                                    ; ERRORS
   4352
   4353 021474  012600                     101$:   MOV     (SP)+,R0        ; GET NODE'S ADDRESS
   4354 021476  012701  000020                     MOV     #C.LGTH,R1      ;  AND LENGTH
   4355 021502                                     CALL    $DEA16          ;   AND DEALLOCATE NODE.
   4356 021506                                     ERRPT$  ERR13,$EROUT    ; TASK ACTIVE
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 4
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

ERR49      003426 R         #9-314      39-2070     40-2092
ERR5       000524 R         #9-226      11-516
ERR6       000560 R         #9-228      11-517
ERR7       000642 R         #9-230      11-520
ERR77      000664 R         #9-232      11-521
ERR8       000714 R         #9-234      39-2056
ERR9       000764 R         #9-236      39-2057     40-2171
EVQUAL     000120 R         #6-142     *56-3130    *56-3133     57-3171    *57-3172    *57-3174     59-3326     59-3332     61-3443
                              61-3445     61-3449     61-3453     63-3690     63-3694    *73-3959    *73-3962    *73-3965    *73-3968
                            *73-3971    *73-3974    *73-3977    *73-3980
EVTVAL     000000 R         #10-325     10-374      46-2639     56-3136
EVTVSZ   = 000020           #10-374     46-2640
EV$ACF   = 000201           #5-79
EV$ADR   = 000420           #5-79
EV$ADU   = 000417           #5-79
EV$APL   = 000400           #5-79
EV$ARC   = 000421           #5-79
EV$AUC   = 000010           #5-79
EV$AUS   = 000003           #5-79
EV$CDF   = 000520           #5-79
EV$COZ   = 000011           #5-79
EV$DBR   = 000302           #5-79
EV$GAS   = 035101           #5-79
EV$HCE   = 035114           #5-79
EV$HCI   = 035113           #5-79
EV$HFE   = 000506           #5-79
EV$IFL   = 000413           #5-79
EV$IFO   = 000415           #5-79
EV$IFS   = 000414           #5-79
EV$INF   = 000515           #5-79
EV$LDL   = 000407           #5-79
EV$LDN   = 010416           #5-79
EV$LDO   = 000411           #5-79
EV$LDS   = 000410           #5-79
EV$LSC   = 000500           #5-79
EV$LUP   = 000412           #5-79
EV$NOL   = 000402           #5-79
EV$NRC   = 000416           #5-79
EV$NSC   = 000200           #5-79
EV$NUL   = 000401           #5-79
EV$NVR   = 000406           #5-79
EV$OPL   = 000403           #5-79
EV$PCC   = 034000           #5-79
EV$PCI   = 034001           #5-79
EV$PCM   = 034002           #5-79
EV$PFE   = 000404           #5-79
EV$PPC   = 034003           #5-79
EV$RCF   = 000517           #5-79
EV$RDC   = 010001           #5-79
EV$RDR   = 010002           #5-79
EV$RJE   = 035106           #5-79
EV$RSC   = 000501           #5-79
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 17
SYMBOL  CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

VF.SS     = ******  GX         39-1947    39-2031    40-2090    40-2108    40-2115    75-4120
V$FLG       000000            *36-1840   *36-1850   *37-1878   *37-1888
V$RCV       000002             36-1839   *36-1851
V$XMT       000012             37-1877   *37-1889
WC.CNT    = 000400            #5-73
WC.EVE    = 004000            #5-73       45-2561
WC.TRI    = 002000            #5-73
WC.UNT    = 001000            #5-73
X25NAM      000004 R          #6-119      40-2097
X5DSMT      020536 R           40-2166   #76-4138
ZF.LLC    = ******  GX         18-824
Z.DAT     = ******  GX         11-442     81-4271    83-4379
Z.FLG     = ******  GX         18-824
Z.LLN     = ******  GX         18-827     74-4007
Z.MAP     = ******  GX         11-432     18-829     35-1803
Z.NAM     = ******  GX         19-865     20-910     21-968     22-1022    81-4266    85-4449
Z.PCB     = ******  GX         83-4377
$ALL15    = ******  GX         39-1987    40-2131    74-4020    82-4306
$ALPOL    = ******  GX         27-1445    28-1554    41-2259    42-2384    59-3312
$BFR      = ******  GX         11-443    *13-597    *13-602    *13-607    *14-640    *15-672     20-922     20-928     21-978
                               21-985     22-1032    22-1038    26-1269    26-1271    26-1273    26-1275   *26-1291   *26-1308
                              *26-1325   *26-1342   *26-1359    27-1389    27-1391    27-1400    27-1404    27-1408    27-1426
                               27-1448    28-1495    28-1509    28-1512   *28-1539    28-1557    28-1574    28-1576    28-1584
                              *29-1613    29-1617    29-1621   *30-1641    30-1645    30-1650   *32-1714    36-1831    36-1839
                              *36-1840   *36-1850   *36-1851    37-1871    37-1877   *37-1878   *37-1888   *37-1889    38-1911
                               38-1913   *38-1915   *38-1918    39-1959    39-1971    39-1991    39-1993    39-1994   *39-1997
                              *39-2001   *39-2002   *39-2005   *39-2006   *39-2007   *39-2008   *39-2009   *39-2012   *39-2013
                              *39-2014    39-2020   *39-2033   *39-2034    39-2035    39-2037    39-2038   *39-2043    39-2045
                               40-2102    40-2118    40-2134    40-2136    40-2137   *40-2139   *40-2143   *40-2144   *40-2147
                              *40-2148   *40-2149   *40-2150   *40-2151   *40-2154   *40-2155   *40-2156    40-2161    41-2215
                              *41-2233   *41-2234   *41-2237   *41-2240   *41-2243   *41-2247   *41-2248   *41-2251   *41-2252
                               41-2262    41-2275    42-2320    42-2321    42-2329    42-2340    42-2346    42-2371    42-2391
                               42-2393   *42-2418    42-2419    42-2428    45-2583    45-2584   *45-2586    45-2587    45-2594
                               45-2601    45-2606    45-2609    45-2610    49-2773    49-2775    49-2777    49-2779    50-2830
                               50-2832    50-2834    50-2836    53-3022    53-3033    54-3067    57-3177    57-3222    59-3319
                               59-3342    59-3343    59-3344   *60-3372    60-3376    60-3380    60-3387    60-3391    60-3393
                               60-3398   *61-3426    61-3430    61-3434    61-3439    61-3443    61-3447    61-3451    61-3455
                               62-3492    62-3510    62-3521    62-3538    62-3544    62-3556    62-3573    62-3575    62-3578
                               62-3587    62-3591    62-3594    62-3600    62-3602    62-3605    62-3608    62-3611    62-3636
                               62-3643    63-3678    63-3708   *74-4011    74-4026   *74-4029   *74-4030    74-4044    75-4092
                              *75-4096    75-4104    75-4108    75-4111   *75-4113   *75-4114   *75-4122    76-4140    76-4142
                              *76-4143   *77-4160    77-4161    77-4165   *78-4183    78-4186   *78-4186   *79-4207   *80-4227
                               81-4272    81-4273    81-4274    81-4278    82-4313    84-4416
$CAT5     = ******  GX         48-2737    50-2868    50-2894
$CEACC    = ******  GX         28-1569    29-1619    30-1648    32-1712    41-2272    42-2422    45-2592    59-3340    60-3378
                               61-3432    62-3502    62-3514    62-3537    62-3577    62-3586    62-3615    62-3635    63-3676
$CMAND    = ******  GX         27-1381    28-1485    36-1834    37-1874    41-2198    42-2302    44-2494    47-2683    57-3185
                               58-3257    58-3268    58-3277    59-3323    62-3482    78-4181
$CNOD       003466 RG         #25-1174
$COTB     = ******  GX         50-2846    50-2850    50-2912    52-2987
$DEA16    = ******  GX         39-1999    39-2066    40-2141    40-2180    75-4116    82-4355
$DECPT    = ******  GX         26-1267    26-1290    26-1307    26-1324    26-1341    26-1358    27-1399    27-1407    28-1573
```

```
VEXP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:27     PAGE 2
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME       REFERENCES

CALL              5-78        5-83        5-86
RETURN            5-88
```

```
                                 ;
                                 ; POINTER TO ALLOCATED SCOM
                                 ;
                                         .ENABL  LSB
001064  112524                   SCOM:   MOVB    (R5)+,(R4)+     ; COPY ADDRESS
001066  112524                           MOVB    (R5)+,(R4)+     ;       ...
001070  112546                           MOVB    (R5)+,-(SP)     ; ASSEMBLE COUNT
001072  112566  000001                   MOVB    (R5)+,1(SP)     ;       ...
001076  012603                           MOV     (SP)+,R3        ;       ...
001100  001407                           BEQ     10$             ; IF ZERO, NO UNLOAD DATA TO STORE
001102  016702  000000G          SAVALC: MOV     $UDA,R2         ; GET UNLOAD DATA ADDRESS
001106  016422  177776                   MOV     -2(R4),(R2)+    ; STORE ALLOCATION ADDRESS
001112  010322                           MOV     R3,(R2)+        ; STORE COUNT
001114  010267  000000G                  MOV     R2,$UDA         ; SET NEW UNLOAD DATA ADDRESS
001120                           10$:    RETURN
                                         .DSABL  LSB
```

```
VEX2       CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

LF.TIM  = 000010            #5-60
LF.UNL  = 020000            #5-60
LF.X2P  = 000000            #5-60
LIBR      001434 R           5-105      #12-576
LINKS     000360 R           5-106      #7-305
LN.CLO  = 000000            #5-60
LN.DUM  = 000005            #5-60
LN.LOA  = 000004            #5-60
LN.LOO  = 000003            #5-60
LN.OAU  = 000003            #5-60
LN.OFF  = 000001            #5-60
LN.ON   = 000000            #5-60
LN.OOP  = 000004            #5-60
LN.OPE  = 000001            #5-60
LN.REF  = 000002            #5-60
LN.SER  = 000002            #5-60
LN.STA  = 000017            #5-60
LN.SUB  = 000360            #5-60
LN.TRI  = 000006            #5-60
LSTHD     000424 R           5-108      #7-315
LTAB      000440 R           5-109      #7-324
L.COST    000015            #5-60
L.CTL     000012            #5-60
L.CVA     177776            #5-60       *7-270      *7-271
L.DDM     000002            #5-60
L.DDS     000004            #5-60
L.DLC     000003            #5-60
L.DLM     000006            #5-60
L.DLS     000010            #5-60
L.FLG     000000            #5-60
L.KRBA    000016            #5-60
L.LEN   = 000022            #5-60
L.MPF     000022            #5-60
L.NMST    000020            #5-60
L.NSTA    000014            #5-60
L.OWNR    000021            #5-60
L.UNT     000013            #5-60
MPTAB     001276 R           5-110      #11-542
MXPTB     001754 R           5-128      #13-713
MXTAB     001204 R           5-111      #11-519
NOD       000570 R           5-112      #7-372
NXTLN     000000 R          #5-78       *12-607     *12-650     12-651
PECHA     000310 R           5-129      #7-284
PRI       001172 R           5-113      #10-512
PRINT     001570 R           10-506     #12-614
PRLT      001516 R           6-178      #12-599
PRTOV     000616 R           5-114      #7-382
PR7     = ******   GX        10-512
R$$MPL  = ******             12-619      12-640     12-668
SAVALC    001102 R          #9-483       13-761
SCOM      001064 R           5-115      #9-477
SECSR     000446 R           5-116      #7-330
```

```
                                    ;
                                    ; SECONDARY LINKED FILE
                                    ;
000202  005002                      XMXPTB: CLR     R2              ; GET REPEAT COUNT
000204  152502                              BISB    (R5)+,R2        ; ...
000206  112546                              MOVB    (R5)+,-(SP)     ; GET MUX TABLE ADDRESS
000210  112566  000001                      MOVB    (R5)+,1(SP)     ; ...
000214  012600                              MOV     (SP)+,R0        ; ...
000216  005702                              TST     R2              ; IS REPEAT COUNT 0 ?
000220  001407                              BEQ     20$             ; IF EQ, YES .. DO NOTHING
000222  005001                              CLR     R1              ; CALCULATE LENGTH OF MUX TABLES
000224  166701  000000G             10$:    SUB     $MXLEN,R1       ; ...
000230  005302                              DEC     R2              ; DO IT <FILE COUNT> TIMES
000232  001374                              BNE     10$             ; ...
000234                                      CALL    $DEA16          ; DEALLOCATE MUX TABLES
000240                              20$:    RETURN


        000001                              .END
```

```
                                        ;
                                        ; COMPLEMENT OF PRIORITY
                                        ;
000606  062703  000002                  PRI:    ADD     #2,R3
000612                                          RETURN

                                        ;
                                        ; LOCAL NODE ADDRESS
                                        ;

000614  005723                          NOD:    TST     (R3)+
000616                                          RETURN

                                        ;
                                        ; PROTOCOL OVERHEAD SIZE
                                        ;

000620  105723                          PRTOV:  TSTB    (R3)+
000622  105725                                  TSTB    (R5)+
000624                                          RETURN

                                        ;
                                        ; POINTER TO ALLOCATED SCOM
                                        ;
000626  062705  000004                  SCOM:   ADD     #4,R5
000632  062703  000002                          ADD     #2,R3
000636                                          RETURN

                                        ;
                                        ; SECONDARY CSR
                                        ;
000640  005767  000154'                 SECSR:  TST     FLAGS           ; IS IT MUX EXPANSION ?
000644  001451                                  BEQ     20$             ; NO ..
000646  016700  000152'                         MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
000652  126400  000013                          CMPB    L.UNT(R4),R0    ; DO UNIT NUMBERS MATCH ?
000656  001044                                  BNE     20$             ; NO ..
000660  032767  000000G 000000G                 BIT     #FL.KMX,$FLAGS  ; IS THIS A COMIOP MUX LINE ?
000666  001403                                  BEQ     10$             ; NO ..
000670  006200                                  ASR     R0              ; YES .. DIVIDE COUNTER BY 8
000672  006200                                  ASR     R0              ; (NUMBER OF LINES PER CONTROLLER)
000674  006200                                  ASR     R0              ;
000676  006300                          10$:    ASL     R0              ; CONVERT TO A WORD INDEX
000700  010377  000000G                         MOV     R3,@$ADRZ       ; VIRTUAL ADDRESS
000704  005077  000000G                         CLR     @$ADRX          ; CLEAR THE BIAS
000710  005767  000000G                         TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM?
000714  100413                                  BMI     5$              ; NO IF MINUS
000716  020327  120000                          CMP     R3,#120000      ; PARTITION LINE TABLE?
000722  103410                                  BLO     5$              ; NO IF LO
000724  010346                                  MOV     R3,-(SP)        ; LINE TALE ADDRESS
000726  042716  160000                          BIC     #160000,(SP)    ; CLEAR THE APR BITS
000732  012677  000000G                         MOV     (SP)+,@$ADRZ    ; NEW VIRTUAL ADDRESS
000736  016777  000000G 000000G                 MOV     $BIAS,@$ADRX    ; AND THE PROCSS BIAS
000744                                  5$:     GETAD   ,#4             ; ACCESS THE LINE TABLE
000756  016067  000000G 000000G                 MOV     $$SCSR(R0),$BFR ; COPY SECONDARY CSR ADDRESS
000764                                          PUTAD                   ; RE-WRITE LINE TABLE
000770  062703  000002                  20$:    ADD     #2,R3
000774                                          RETURN
```

```
VEX5      CREATED BY  MACRO  ON 29-JUN-85 AT 02:28     PAGE 4

MACRO CROSS REFERENCE                                  CREF   04.00

MACRO NAME          REFERENCES

CALL                  6-180      6-191      7-290      7-298      7-319      7-321      7-423      7-425      8-490      8-518
                      9-542      9-560     10-632
CRASH                 6-198      7-328      8-503      8-508
GETAD                #5-60       7-290      7-319      7-423      9-542     10-632
GETRV                #5-60
PUTAD                #5-60       7-298      7-321      7-425
PUTRC                #5-60
RAND                 #7-290      7-290     #7-298      7-298     #7-319      7-319     #7-321      7-321     #7-423      7-423
                     #7-425      7-425     #9-542      9-542    #10-632     10-632
RESRG                #5-60       7-295
RETURN                6-193      7-215      7-219      7-228      7-239      7-246      7-258      7-265      7-271      7-300
                      7-323      7-339      7-351      7-357      7-363      7-369      7-375      7-382      7-390      7-397
                      7-427      7-433      7-439      7-445      7-451      7-457      7-463      7-470      8-498      8-526
                      9-569      9-595      9-603      9-609     10-674     11-682     11-687     11-692     11-697     11-702
                     11-707     11-712     11-717     11-722     11-727     11-732     12-739     12-744     12-749     12-754
                     12-759
SAVRG                #5-60       7-291
SLTDF$               #5-60       5-65
SWTCHI               #5-60      10-627     10-673
TTAB$                #5-70       5-83       5-84       5-85       5-86       5-87       5-88       5-89       5-90       5-91
                      5-92       5-93       5-94       5-95       5-96       5-97       5-98       5-99       5-100      5-101
                      5-102      5-103      5-104      5-105      5-106      5-107      5-108      5-109      5-110      5-111
                      5-112      5-113      5-114      5-115      5-116      5-117      5-118      5-119      5-120      5-121
                      5-122      5-123      5-124      5-125      5-126      5-127      5-128      5-129      5-130      5-131
                      5-132      5-133      5-134      5-135
```

```

                                    ;
                                    ; NETWORK NAME
                                    ;
 000350  016700  000000G            DTENET: MOV     .PSTCN,R0               ; GET SIZE OF SUPPLIED NAME
 000354  020027  000006                     CMP     R0,#DTNMMX              ; BIGGER THAN MAXIMUM?
 000360  101010                             BHI     101$                    ; BR IF YES, ERROR
 000362  016701  000000G                    MOV     .PSTPT,R1               ; POINT TO NAME
 000366  012702  000027'                    MOV     #DTNTNM,R2              ; POINT TO LOCAL COPY
 000372  112122                     10$:    MOVB    (R1)+,(R2)+             ; COPY NAME
 000374  005300                             DEC     R0                      ; MORE TO COPY?
 000376  003375                             BGT     10$                     ; BR IF YES
 000400  000403                             BR      DTEEND                  ; ELSE MERGE INTO END PROCESSING

                                    ;
                                    ; ERRORS
                                    ;
 000402                             101$:   EMSG$R  00                      ; ILLEGAL NETWORK NAME PARAMETER

```

```
VFGDTE     CREATED BY  MACRO  ON 29-JUN-85 AT 02:28     PAGE 5

MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME         REFERENCES

ASL$               #5-61     20-508
CALL                9-158    12-256    12-264    12-278    12-282    12-288    15-353    16-377    16-380    16-388
                    16-389   18-432    19-475
CEACC$             #5-60     16-388
DBGTP$             #10-173  #10-185   #10-204   #10-208   #10-212   #10-219   #11-237
DTEDF$             #5-60     5-65
EMSG$R             #5-60     9-167     12-269    13-316    13-327    14-347    15-359    18-452
GETAD              #5-60     16-389
GETRV              #5-61     12-288    16-380
ISTAT$             #5-61     10-173
MTRAN$             #10-173
NTLER$             #5-61     8-128     8-129     8-130     8-131     8-132
PHBDF$             #5-61     5-66
RAND               #12-288   12-288   #16-380    16-380   #16-389    16-389
RESMAP             #5-75     16-399
RESRG              #5-60
RETURN              9-162    12-257    12-265    12-296    13-312    13-323    15-355    16-401    17-422    18-434
                    18-441   18-448    19-479    20-516
SAVMAP             #5-71     16-381
SAVRG              #5-60
SLTDF$             #5-60     5-67
STATE$             #5-61     10-177   #10-180   #10-183   #10-187   #10-190   #10-193   #10-196   #10-199   #10-202
                   #10-206  #10-210   #10-214   #10-217    11-226   #11-229   #11-232   #11-235   #11-239   #11-242
TRAN$              #5-61    #10-178   #10-181   #10-184   #10-185   #10-188   #10-191   #10-194   #10-197   #10-200
                   #10-203  #10-204   #10-207   #10-208   #10-211   #10-212   #10-215   #10-218   #10-219   #11-227
                   #11-230  #11-233   #11-236   #11-237   #11-240
```

```


                                        ;
                                        ; END OF PVC$DF MACRO
                                        ;
 000346  016701  000026'                PVEND:  MOV     DLMSTA,R1           ; GET STATION NUMBER
 000352  006301                                 ASL     R1                  ; MAKE IT A WORD INDEX
 000354  016167  000000G 000000'                MOV     $$HTIM(R1),HTIMR;   GET HELLO TIMER
 000362  016167  000000G 000002'                MOV     $$LTIM(R1),LTIMR;   AND LISTEN TIMER

 000370                                         CALL    $SAVAL              ; SAVE ALL REGISTERS
 000374                                         SAVMAP                      ; SAVE PREVIOUS MAPPING
 000400                                         CALL    FNDPRT              ; FIND FREE PORT ENTRY
 000404  103410                                 BCS     102$                ; BR IF ONE NOT FOUND
 000406                                         CALL    DLMTLC              ; ELSE STORE PORT NUMBER IN COUNTER BLOCK
 000412                                         CALL    ALLXCB              ; ALLOCATE X25 CIRCUIT BLOCK
 000416  103403                                 BCS     102$                ; BR IF UNSUCCESSFUL
 000420                                         RESMAP                      ; RESTORE PREVIOUS MAPPING
 000424                                         RETURN
                                        ;
                                        ; ERRORS
                                        ;
 000426                                 102$:   EMSG$R  N3                  ; PVC/XCB/PORT/HASH ALLOCATION FAILED
```

```
VFGPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:30      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE             REFERENCES

INVCIR     001546 R          19-649      19-653     #19-660      19-667      19-669      19-681      19-683
LF.ACT   = 100000           #5-70
LF.BRO   = 000400           #5-70
LF.BWT   = 000007           #5-70
LF.ENA   = 002000           #5-70
LF.LPB   = 001000           #5-70
LF.MDC   = 000100           #5-70
LF.MFL   = 004000           #5-70
LF.MTP   = 000020           #5-70
LF.PAC   = 000200           #5-70
LF.RDY   = 040000           #5-70
LF.REA   = 010000           #5-70
LF.SER   = 000040           #5-70
LF.TIM   = 000010           #5-70
LF.UNL   = 020000           #5-70
LF.X2P   = 000000           #5-70
LN.CLO   = 000000           #5-70
LN.DUM   = 000005           #5-70
LN.LOA   = 000004           #5-70
LN.LOO   = 000003           #5-70
LN.OAU   = 000003           #5-70
LN.OFF   = 000001           #5-70
LN.ON    = 000000           #5-70
LN.OOP   = 000004           #5-70
LN.OPE   = 000001           #5-70
LN.REF   = 000002           #5-70
LN.SER   = 000002           #5-70
LN.STA   = 000017           #5-70
LN.SUB   = 000360           #5-70
LN.TRI   = 000006           #5-70
LTIMR      000002 R         #7-132      *13-407      20-748
L$CHTB     000030             17-588
L$LEN      000122             17-587
L.COST     000015           #5-70
L.CTL      000012           #5-70        19-671
L.CVA      177776           #5-70
L.DDM      000002           #5-70
L.DDS      000004           #5-70
L.DLC      000003           #5-70
L.DLM      000006           #5-70
L.DLS      000010           #5-70
L.FLG      000000           #5-70
L.KRBA     000016           #5-70
L.LEN    = 000022           #5-70
L.MPF      000022           #5-70
L.NMST     000020           #5-70
L.NSTA     000014           #5-70
L.OWNR     000021           #5-70
L.UNT      000013           #5-70
MAXID    = 000020           #6-106       7-149
N$TLC      000100             20-732      20-733       20-734
N$XLEN     000124             20-731
```

```
                                                .SBTTL  SVC$DF STATE TABLE
                                        ;
                                        ; TPARS STATE TABLES
                                        ;
 000052                                         ISTAT$  SVCST,SVCKW
                                        ;
                                        ; SVC$DF
                                        ;
 000052                                         STATE$  SVCDF
 000052                                          TRAN$   %SVC$DF%,,,1,SYNERR

 000052                                         STATE$
 000052                                          TRAN$   $RAD50,,SVNAM  ; PROCESS NAME

 000052                                         STATE$
 000052                                          TRAN$   '-

 000052                                         STATE$
 000052                                          TRAN$   $DNUMB,,SVCTL  ; CONTROLLER NUMBER

 000052                                         STATE$
 000052                                          TRAN$   <'.>

 000052                                         STATE$
 000052                                          TRAN$   $DNUMB,,SVSTA  ; STATION NUMBER

 000052                                         STATE$
 000052                                          TRAN$   <',>

 000052                                         STATE$                  ; PACKET SIZE
 000052                                          TRAN$   <',>,SVCWND
 000052                                          TRAN$   $NUMBR,,SVPKT

 000052                                         STATE$
 000052                                          TRAN$   <',>

 000052                                         STATE$  SVCWND          ; WINDOW SIZE
 000052                                          TRAN$  <',>,SVCFLG
 000052                                          TRAN$  $NUMBR,,SVWND

 000052                                         STATE$
 000052                                          TRAN$   <',>

 000052                                         STATE$  SVCFLG          ; FLAGS
 000052                                          TRAN$  $NUMBR,,SVFLAG

 000052                                         STATE$
 000052                                          TRAN$   <',>

 000052                                         STATE$                  ; MAX RECALLS
 000052                                          TRAN$   $NUMBR,,SVRCLL

 000052                                         STATE$
 000052                                          TRAN$   <',>

 000052                                         STATE$                  ; RECALL TIMER
 000052                                          TRAN$   $NUMBR,,SVRRTM
```

```
  3519 016170   001056                              BNE     15$                             ; BR IF NO
  3520 016172                                       SWTCHO  #FLTBUF                         ; USE FLTBUF AS OUTPUT BUFFER
  3521 016200   016767  000000G 000170'             MOV     $BFR,FLTBUF                     ; STORE POINTER TO NEXT BLOCK
  3522 016206   010001                              MOV     R0,R1                           ; SAVE ADDRESS OF PREVIOUS BLOCK
  3523 016210                              12$:     CALL    GETNXT                          ; GET NEXT FILTER BLOCK IN LIST
  3524 016214   103044                              BCC     15$                             ; BR IF FINISHED
  3525 016216   012703  000106'                     MOV     #GMASK,R3                       ; POINT TO GLOBAL FILTER MASKS
  3526 016222   012705  000174'                     MOV     #FLTBUF+F.SEV,R5                ; POINT TO 'SET' FILTER MASKS
  3527 016226                                       CALL    CLREVT                          ; CLEAR ANY EVENTS SET IN GLOBAL MASK
  3528 016232   103013                              BCC     14$                             ; BR IF SOME MASKS STILL SET
  3529 016234   012705  000230'                     MOV     #FLTBUF+F.CEV+10,R5             ; POINT TO END OF 'CLEAR' FILTER MASKS
  3530 016240                                       CALL    CHKEVT                          ; ANY 'CLEAR' FILTER MASKS SET?
  3531 016244   103006                              BCC     14$                             ; BR IF YES
  3532 016246   012705  000170'                     MOV     #FLTBUF,R5                      ; DEALLOCATE FILTER BLOCK FROM FLTBUF
  3533 016252   010400                              MOV     R4,R0                           ; GET ADDRESS OF FILTER BLOCK
  3534 016254                                       CALL    DEAFIL                          ; DEALLOCATE FILTER BLOCK
  3535 016260   000753                              BR      12$                             ; CHECK NEXT BLOCK
  3536 016262                              14$:     SAVRG   <R4>                            ; SAVE UNMAPPED ADDRESS
  3537 016264                                       CEACC$  R4                              ; CONVERT TO MAPPED ADDRESS
  3538 016274   016767  000002G 000000G             MOV     F.CLS+$BFR,$RSIZE               ; GET LENGTH OF BLOCK TO DEALLOCATE
  3539 016302   042767  177700  000000G             BIC     #^C<FF.MSK>,$RSIZE              ; ...
  3540 016310                                       PUTAD   R4                              ; REWRITE FILTER BLOCK
  3541 016320                                       RESRG   <R4>                            ; RESTORE UNMAPPED ADDRESS
  3542 016322   010401                              MOV     R4,R1                           ; SAVE ADDRESS OF PREVIOUS
  3543 016324   000731                              BR      12$                             ; GET NEXT BLOCK IN LIST
  3544 016326                              15$:     SWTCHO                                  ; USE $BFR AS OUTPUT BUFFER
  3545 016334   000241                              CLC                                     ; FILTER BLOCK STILL USEFUL
  3546 016336   000167  000526                      JMP     80$                             ; FINISH UP
  3547                                     ;
  3548                                     ; CLEAR EVENT FILTER OPERATION
  3549                                     ;
  3550 016342   020002                     20$:     CMP     R0,R2                           ; OPERATE ON GLOBAL FILTER BLOCK?
  3551 016344   001526                              BEQ     60$                             ; BR IF YES
  3552 016346   012703  000004G                     MOV     #LO.EVT+RQB,R3                  ; POINT TO FILTER MASKS
  3553 016352   012705  000076'                     MOV     #TMPMSK,R5                      ; POINT TO TEMPORARY SAVE AREA
  3554 016356                                       CALL    SAVFIL                          ; SAVE FILTER MASKS
  3555 016362   012703  000004G                     MOV     #LO.EVT+RQB,R3                  ; POINT TO FILTER MASKS
  3556 016366   012705  000004G                     MOV     #$BFR+F.SEV,R5                  ; POINT TO 'SET' FILTER MASKS
  3557 016372                                       CALL    CLREVT                          ; CLEAR THE 'SET' EVENT MASKS
  3558
  3559 016376   012703  000076'                     MOV     #TMPMSK,R3                      ; POINT TO SAVE AREA
  3560 016402   012705  000004G                     MOV     #LO.EVT+RQB,R5                  ; POINT TO FILTER MASKS
  3561 016406                                       CALL    SAVFIL                          ; RESTORE FILTER MASKS
  3562 016412   005702                              TST     R2                              ; IS THERE A GLOBAL FILTER BLOCK?
  3563 016414   001450                              BEQ     40$                             ; BR IF NO
  3564 016416   012703  000106'                     MOV     #GMASK,R3                       ; POINT TO GLOBAL FILTER MASKS
  3565 016422   012705  000076'                     MOV     #TMPMSK,R5                      ; POINT TO TEMPORARY STORAGE
  3566 016426                                       CALL    SAVCOM                          ; SAVE COMPLEMENT OF GLOBAL MASKS
  3567 016432   012705  000004G                     MOV     #RQB+LO.EVT,R5                  ; POINT TO FILTER MASKS
  3568 016436   012703  000076'                     MOV     #TMPMSK,R3                      ; POINT TO COMPLEMENTED FILTER MASKS
  3569 016442                                       CALL    CLREVT                          ; DETERMINE IF ANY 'CLEAR' MASKS TO SET
  3570 016446   103406                              BCS     30$                             ; BR IF NONE TO SET
  3571
  3572 016450   012703  000004G                     MOV     #LO.EVT+RQB,R3                  ; POINT TO FILTER MASKS
  3573 016454   012705  000030G                     MOV     #$BFR+F.CEV,R5                  ; POINT TO 'CLEAR' FILTER MASKS
  3574 016460                                       CALL    SETEVT                          ; SET THE 'CLEAR' FILTER MASKS
  3575 016464                              30$:     SWTCHO  #$BFR                           ; POINT TO FILTER BLOCK
```

```
                                        .SBTTL  DEAFIL - DEALLOCATE FILTER BLOCK

                                ;+
                                ;
                                ; DEAFIL - DEALLOCATE FILTER BLOCK
                                ;
                                ; INPUTS:
                                ;       R0 - UNMAPPED ADDRESS OF BLOCK TO DEALLOCATE
                                ;       R1 - UNMAPPED ADDRESS OF PREVIOUS BLOCK IN LIST
                                ;       R5 - ADDRESS OF BUFFER CONTAINING FILTER BLOCK TO DEALLOCATE
                                ;
                                ;-

017600  010146                  DEAFIL: MOV     R1,-(SP)                ; SAVE R1
017602  016546  000002                  MOV     F.CLS(R5),-(SP)         ; GET CLASS AND SIZE WORD
017606  042716  177700                  BIC     #^C<FF.MSK>,(SP)        ; ISOLATE FILTER BLOCK SIZE
017612  020127  000000G                 CMP     R1,#$FILHD              ; IS THE PREVIOUS THE FIRST ONE?
017616  001003                          BNE     10$                     ; BR IF NO
017620  011567  000000G                 MOV     (R5),$FILHD             ; ELSE UPDATE LISTHEAD
017624  000404                          BR      20$                     ; AND DEALLOCATE THE BLOCK
017626  010146                  10$:    MOV     R1,-(SP)                ; GET UNMAPPED ADDR OF PREVIOUS BLOCK
017630  010046                          MOV     R0,-(SP)                ; GET UNMAPPED ADDRESS OF CURRENT BLOCK
017632                                  CALL    $XLINK                  ; UNLINK FROM LIST
017636  012601                  20$:    MOV     (SP)+,R1                ; GET LENGTH OF BLOCK TO DEALLOCATE
017640                                  CALL    $DEPOL                  ; DEALLOCATE FILTER BLOCK
017644  012601                          MOV     (SP)+,R1                ; RESTORE R1
017646                                  RETURN
```

```
                                                .SBTTL  CHKXPT - CHECK XPT CONTEXT
                                        ;+
                                        ; **-CHKXPT-CHECK XPT CONTEXT
                                        ;
                                        ;       MAKE SURE TRANSPORT(XPT) IS LOADED AND IT'S DATA BASE IS ALLOCATED.
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R0      - PDV ADDRESS
                                        ;       R1      - XPT PDV INDEX
                                        ;       R2      - XPT DATABASE VIRTUAL ADDRESS
                                        ;-

021516                                  CHKXPT:
021516  012700  114224                          MOV     #^RXPT,R0               ; XPT IN RAD50
021522                                          CALL    $FPDV                   ; FIND PDV ADDRESS
021526  103407                                  BCS     101$                    ; IF CS, NO XPT
021530  005760  000000G                         TST     Z.PCB(R0)               ; IS PROCESS LOADED ?
021534  001404                                  BEQ     101$                    ; NO
021536  016002  000000G                         MOV     Z.DAT(R0),R2            ; GET DATABASE ADDRSS
021542  001416                                  BEQ     102$                    ; DATABASE NOT ALLOCATED
021544                                          RETURN


                                        ;
                                        ; ERRORS
                                        ;

021546  032767  000000C 000000G 101$:   BIT     #VF.PEM!VF.DLX,$VFLAG   ; IS THIS A PROTOCOL EMULATOR COMMAND
                                                                        ;  OR DLX ONLY SYSTEM?
021554  001023                                  BNE     103$                    ; IF NE, YES .. NO ERROR MESSAGE PLEASE
021556  012700  015370                          MOV     #^RDLX,R0               ; SURE NOT DLX ONLY (BIT NOT YET SET)?
021562                                          CALL    $FPDV                   ; LOOK FOR DLX PDV
021566  103016                                  BCC     103$                    ; BR IF FOUND - ASSUME DLX ONLY
021570                                          ERRPT$  ERR3,$EROUT             ; XPT NOT LOADED
021600  032767  000000G 000000G 102$:   BIT     #VF.PEM,$VFLAG          ; IS THIS A PROTOCOL EMULATOR COMMAND ?
021606  001006                                  BNE     103$                    ; IF NE, YES .. NO ERROR MESSAGE PLEASE
021610                                          ERRPT$  ERR4,$EROUT,FATAL       ; XPT DATABASE NOT ALLOCATED
021624  000261                          103$:   SEC                             ; JUST INDICATE FAILURE
021626                                          RETURN
```

```
/ENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08     PAGE 5
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE               REFERENCES

EV$PUL  = 000405            #5-79
EV$SNA  = 035000            #5-79
EV$SNF  = 000516            #5-79
EV$SPE  = 035001            #5-79
EV$XCE  = 034110            #5-79
EV$XDI  = 013600            #5-79
EV$XGW  = 034111            #5-79
EV$XMX  = 000514            #5-79
EV$XRS  = 000512            #5-79
EV$XSC  = 000513            #5-79
EV$X2S  = 013500            #5-79
EV.CCB  = 000001            #5-79
EV.CIR  = 000020            #5-79
EV.LCB  = 000100            #5-79
EV.LIN  = 000004            #5-79
EV.MAP  = 000002            #5-79
EV.MOD  = 000040            #5-79
EV.NOD  = 000010            #5-79
EV.PRT  = 000200            #5-79
E$DATA    000020            #5-79
E$EVTS    000000            #5-79
E$LCN     000016            #5-79
E$LIN     000000            #5-79
E$MOD     000012            #5-79
E$NBR     000014            #5-68
E$NBS     000020            #5-68
E$NCR     000034            #5-68
E$NCS     000036            #5-68
E$NIC     000044            #5-68
E$NLEN    000050            #5-68
E$NLLA    000012            #5-68
E$NLNK    000000            #5-68
E$NML     000040            #5-68
E$NMR     000024            #5-68
E$NMS     000030            #5-68
E$NNOD    000002            #5-68
E$NOD     000010            #5-79
E$NRT     000042            #5-68
E$NRTP    000005            #5-68
E$NSEG    000010            #5-68
E$NTIM    000046            #5-68
E$NUSE    000004            #5-68
E$PORT    000014            #5-79
E$PRM     000002            #5-79
E$STAT    000006            #5-79
E$STRT    000006            #5-68
E$TCB     000004            #5-79
E.CTL     000020            #5-79
E.DATA    000046            #5-79
E.EVT     000002            #5-79
E.LCN     000042            #5-79
E.LEN     000216            #5-79
```

```
VENA      CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 18
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE            REFERENCES

                           28-1583   42-2334   42-2339   42-2345   42-2390   42-2427   53-3020   74-4009   74-4025
                           78-4179
$DEPOL   = ******  GX      27-1416   41-2222   42-2356   72-3940
$DEVHD   = ******  GX      77-4160
$EROUT   = ******  GX      11-514    11-515    11-516    11-517    11-520    11-521    11-522    13-614    13-615
                           14-646    14-647    15-678    15-679    16-752    16-753    16-754    16-755    16-756
                           17-802    17-803    20-940    20-941    21-995    21-996    22-1048   22-1049   24-1154
                           25-1217   27-1465   27-1466   27-1467   28-1591   28-1592   28-1593   28-1594   38-1925
                           38-1926   39-2056   39-2057   39-2058   39-2062   39-2063   39-2064   39-2065   39-2069
                           39-2070   40-2092   40-2170   40-2171   40-2172   40-2176   40-2177   40-2178   40-2179
                           41-2283   41-2284   41-2285   42-2313   42-2363   42-2436   42-2437   42-2438   43-2469
                           44-2534   46-2664   47-2712   47-2713   53-3042   54-3073   57-3228   59-3353   79-4210
                           80-4230   82-4356   83-4393   83-4396
$FILHD   = ******  GX      45-2585   59-3346   60-3371   61-3425   72-3932  *72-3934
$FNADD   = ******  GX      28-1499   28-1531   42-2368
$FNALI   = ******  GX      42-2315   42-2374   43-2471
$FNNAM   = ******  GX      27-1386   27-1421   28-1493   28-1507   28-1527   28-1536   53-3029   54-3065
$FNOBJ   = ******  GX      41-2208   41-2229
$FPDV      021716  RG      11-401    11-407    11-440    40-2086   44-2491   83-4375   83-4391  #85-4442
$GETAD   = ******  GX      20-919    20-927    21-977    21-984    22-1031   22-1037   29-1620   30-1649   32-1713
                           39-2040   45-2593   45-2599   60-3379   60-3384   61-3433   61-3438   63-3677   63-3682
                           81-4275
$GETRV   = ******  GX      11-442    26-1267   26-1290   26-1307   26-1324   26-1341   26-1358   27-1399   28-1573
                           36-1830   36-1832   37-1870   37-1872   38-1912   39-1990   39-1993   40-2133   40-2136
                           42-2339   42-2390   49-2774   50-2831   53-3020   74-4009   74-4025   74-4028   75-4094
                           75-4103   76-4139   76-4142   77-4164   78-4179   81-4271   82-4321   82-4328   82-4333
                           82-4344   84-4415   84-4422
$HMBUF   = ******  GX      27-1398   27-1402  *27-1404   27-1406   28-1572   28-1577   28-1579   28-1582   42-2338
                          *42-2342   42-2344   42-2389   42-2418  *42-2424   42-2426   53-3019   53-3021
$LGCON   = ******  GX     *58-3262  *58-3263
$LGFNB   = ******  GX      58-3247   58-3283
$LGMON   = ******  GX     *58-3273  *58-3274
$LGSTT   = ******  GX     *55-3102  *55-3104
$LLCTA   = ******  GX      11-418    16-719    16-740
$LSTAD   = ******  GX      31-1675   31-1680  *31-1682
$LSTRM   = ******  GX      31-1683  *31-1684
$MUL     = ******  GX      74-4016   75-4107
$OBJHD   = ******  GX      29-1612   41-2213  *41-2215   41-2277
$OPTON   = ******  GX      25-1181   25-1187   25-1189   25-1191   25-1193   25-1196   41-2231   41-2238   41-2244
                           41-2249   42-2311   42-2361   43-2467   44-2496   44-2498   44-2510   44-2523   45-2558
                           47-2696   47-2701   55-3093   55-3096   56-3134   56-3148   57-3187   57-3192   58-3255
                           58-3266   59-3308   59-3333   71-3892   71-3897   71-3905   73-3963   73-3966   73-3969
                           73-3972   73-3975   73-3978
$PDVNM   = ******  GX      19-862    85-4445
$PDVTA   = ******  GX      18-822    19-861    19-872    20-908    21-966    22-1020   81-4264   85-4444   85-4457
$PUTAD   = ******  GX      13-608    14-641    15-673    27-1434   28-1518   28-1540   28-1570   32-1715   39-2046
                           41-2253   41-2273   42-2423   45-2620   59-3341   62-3503   62-3515   62-3540   62-3580
                           62-3589   62-3616   62-3638   79-4208   80-4228
$PUTRC   = ******  GX      11-509    26-1292   26-1309   26-1326   26-1343   26-1360   27-1407   28-1583   36-1853
                           37-1891   38-1919   39-1998   39-2003   39-2015   39-2044   40-2140   40-2145   40-2157
                           42-2345   42-2427   74-4013   74-4031   74-4065   75-4097   75-4115   75-4123   76-4144
                           78-4188   82-4346   82-4348
```

```
**FILE**ID**VEX2





VV      VV   EEEEEEEEEE   XX      XX     222222
VV      VV   EEEEEEEEEE   XX      XX     222222
VV      VV   EE           XX      XX   22      22
VV      VV   EE           XX      XX   22      22
VV      VV   EE             XX  XX             22
VV      VV   EE             XX  XX             22
VV      VV   EEEEEEEE         XX             22
VV      VV   EEEEEEEE         XX             22
VV      VV   EE             XX  XX         22
VV      VV   EE             XX  XX         22
  VV  VV     EE           XX      XX     22             ....
  VV  VV     EE           XX      XX     22             ....
    VV       EEEEEEEEEE   XX      XX   2222222222       ....
    VV       EEEEEEEEEE   XX      XX   2222222222       ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                ;
                                ; INTERRUFT ENTRY POINT
                                ;
001122                          INT:
001122  005767  000000G                 TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM ?
001126  100412                          BMI     5$              ; NO

001130  012724  013746                  MOV     #13746,(R4)+    ; STORE "MOV @#KISAR5,-(SP)"
001134  012724  172352                  MOV     #172352,(R4)+   ;       ...
001140  012724  012737                  MOV     #12737,(R4)+    ; STORE "MOV BIAS,@#KISAR5"
001144  016724  000000G                 MOV     $BIAS,(R4)+     ;       ...
001150  012724  172352                  MOV     #172352,(R4)+   ;       ...

001154  012724  004537          5$:     MOV     #4537,(R4)+     ; STORE "JSR R5,@#ENTRY"
001160  112524                          MOVB    (R5)+,(R4)+     ;       ...
001162  112524                          MOVB    (R5)+,(R4)+     ;       ...
001164                                  CALL    PRINT           ; SET UP ICB
001170                                  RETURN

                                ;
                                ; COMPLEMENT OF PRIORITY
                                ;
001172  012714  000000G         PRI:    MOV     #PR7,(R4)       ; STORE PRIORITY 7
001176  046724  000000G                 BIC     $$PRI,(R4)+     ; CLEAR ACTUAL PRIORITY BITS
001202                                  RETURN
```

```
VEX2      CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 3
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE           REFERENCES

SF.ACT  = 000200          #5-60
SF.ENA  = 000100          #5-60
SF.LPB  = 000004          #5-60
SF.MFL  = 000040          #5-60
SF.PAC  = 000020          #5-60
SF.REA  = 000010          #5-60
SF.SER  = 000001          #5-60
SF.SVC  = 000002          #5-60
SF.UNL  = 000040          #5-60
SLNB      000514 R         5-117     #7-347
SLNW      000540 R         5-118     #7-358
STNB      000622 R         5-119     #7-389
STNW      000660 R         5-120     #7-402
SYSFDB  = ******   GX     *7-305     *7-306     *7-309     *7-373     *12-667    *12-670    *13-747    *13-748    *13-755
S.COST    000001          #5-60
S.FLG     000000          #5-60
S.LEN     000004          #5-60
S.NMST    000002          #5-60
S.OWNR    000003          #5-60
TABLE     000274 R        #5-93       6-187     11-528     11-554     13-730
TIME      002206 R         5-127     #13-766
UNB       000722 R         5-121     #7-417
UNW       000744 R         5-122     #7-427
WORD      000766 R         5-123     #7-437
WRVBL     001714 R         6-190     #12-665
X2CH01    002342 R         5-141     #15-847
X2CH02    002350 R         5-142     #15-852
X2CH03    002356 R         5-143     #15-857
X2CH04    002364 R         5-144     #15-862
X2CH05    002372 R         5-145     #15-867
X3CH01    002240 R         5-130     #14-790
X3CH02    002246 R         5-131     #14-795
X3CH03    002254 R         5-132     #14-800
X3CH04    002262 R         5-133     #14-805
X3CH05    002270 R         5-134     #14-810
X3CH06    002276 R         5-135     #14-815
X3CH07    002304 R         5-136     #14-820
X3CH08    002312 R         5-137     #14-825
X3CH09    002320 R         5-138     #14-830
X3CH10    002326 R         5-139     #14-835
X3CH11    002334 R         5-140     #14-840
$ADDR   = ******   GX      6-180
$BFR    = ******   GX      7-309      7-375     13-755
$BIAS   = ******   GX      10-500     12-637
$BIN    = ******   GX      6-179
$BUFCK  = ******   GX      6-188      11-529
$CTIM   = ******   GX      7-251
$DECPT  = ******   GX      7-306      7-373
$FLAGS  = ******   GX      7-258      7-260      7-285      7-287      7-333
$GETLB  = ******   GX      12-583
$GETRV  = ******   GX      7-306      7-373
$HMBUF  = ******   GX      7-305      7-307
```

```
VEX3 - VNP EXPANSION ROUTINES C MACRO V05.03b Saturday 29-Jun-85 02:27  Page 10-1
Symbol table

A$$CHK= 000000              G$$TSS= 000000              M$$CRX= 000000              POP2    000044R             XCORE   000106R
A$$CPS= 000000              G$$TTK= 000000              M$$FCS= 000000              POP3    000036R             XLIBR   000146R
A$$PRI= 000000              G$$WRD= 000000              M$$MGE= 000000              P$$P45= 000000              XMXPTB  000202R
A$$TRP= 000000              I$$RAR= 000000              M$$NET= 000000              P$$WRD= 000000              XSCOM   000056R
C$$CKP= 000000              I$$RDN= 000000              M$$OVR= 000000              Q$$OPT= 000010              X$$DBT= 000000
C$$ORE= 000400              K$$CNT= 177546              NULL    000054R             R$$DER= 000000              $ADDR = ****** GX
C$$RSH= 177564              K$$CSR= 177546              N$$ACC= 000001              R$$K11= 000001              $BIN  = ****** GX
D$$BUG= 177514              K$$LDC= 000000              N$$BUF= 000001              R$$SND= 000000              $DEA16= ****** GX
D$$ISK= 000000              K$$TPS= 000074              N$$LDV= 000001              R$$11M= 000000              $DEA18= ****** GX
D$$L11= 000001              LD$LP = 000000              N$$MCP= 000001              S$$WRG= 000000              $MXLEN= ****** GX
D$$YNC= 000000              L$$ASG= 000000              N$$MLL= 000001              S$$YSZ= 007600              $PTR  = ****** GX
D$$YNM= 000000              L$$DRV= 000000              N$$MOV= 000010              TABLE   000000R        002  $VEX3   000000RG
E$$XPR= 000000              L$$P11= 000001              N$$NCT= 000001              T$$KMG= 000000              $XLBR = ****** GX
F$$LVL= 000001              L$$11R= 000000              N$$PEM= 000001              T$$MIN= 000000              .MGMGE= ****** GX
G$$TPP= 000000              M$$CRB= 000124              POP1    000050R             V$$CTR= 001000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000242    001   (RW,I,LCL,REL,CON)
DATA    000152    002   (RW,D,LCL,REL,CDN)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8955 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:08.30
SY:VEX3.V2,[132,134]VEX3/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VEX3
```

```

                                     ;
                                     ; SYSTEM LINE NUMBER (BYTE)
                                     ;
 000776  005203                      SLNB:   INC     R3
 001000                                      RETURN

                                     ;
                                     ; SYSTEM LINE NUMBER (WORD)
                                     ;
 001002  062703  000002              SLNW:   ADD     #2,R3
 001006                                      RETURN

                                     ;
                                     ; STATION NUMBER (BYTE)
                                     ;
 001010  005203                      STNB:   INC     R3
 001012                                      RETURN

                                     ;
                                     ; STATION NUMBER (WORD)
                                     ;
 001014  062703  000002              STNW:   ADD     #2,R3
 001020                                      RETURN

                                     ;
                                     ; UNIT NUMBER (BYTE)
                                     ;
 001022  005203                      UNB:    INC     R3
 001024                                      RETURN

                                     ;
                                     ; UNIT NUMBER (WORD)
                                     ;
 001026  062703  000002              UNW:    ADD     #2,R3
 001032                                      RETURN

                                     ;
                                     ; SINGLE WORD
                                     ;
 001034  122525                      WORD:   CMPB    (R5)+,(R5)+
 001036  062703  000002                      ADD     #2,R3
 001042                                      RETURN
```

```
**FILE**ID**VFGDTE




VV      VV   FFFFFFFFFF     GGGGGGGG   DDDDDDDD     TTTTTTTTTT   EEEEEEEEEE
VV      VV   FFFFFFFFFF     GGGGGGGG   DDDDDDDD     TTTTTTTTTT   EEEEEEEEEE
VV      VV   FF           GG           DD      DD       TT       EE
VV      VV   FF           GG           DD      DD       TT       EE
VV      VV   FF           GG           DD      DD       TT       EE
VV      VV   FF           GG           DD      DD       TT       EE
VV      VV   FFFFFFFF     GG           DD      DD       TT       EEEEEEEE
VV      VV   FFFFFFFF     GG           DD      DD       TT       EEEEEEEE
VV      VV   FF           GG  GGGGGG   DD      DD       TT       EE
VV      VV   FF           GG  GGGGGG   DD      DD       TT       EE
  VV  VV     FF           GG      GG   DD      DD       TT       EE               ....
  VV  VV     FF           GG      GG   DD      DD       TT       EE               ....
    VV       FF             GGGGGG     DDDDDDDD         TT       EEEEEEEEEE       ....
    VV       FF             GGGGGG     DDDDDDDD         TT       EEEEEEEEEE       ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
    350                                     ;
    351                                     ; END OF DTE$DF MACRO
    352                                     ;
    353 000410                              DTEEND: CALL    CHKDTE                  ; CHECK FOR VALID DTE ADDRESS
    354 000414  103401                              BCS     101$                    ; ERROR - DTE ADDRESS NOT FOUND
    355 000416                                      RETURN
    356                                     ;
    357                                     ; ERRORS
    358                                     ;
    359 000420                              101$:   EMSG$R  03                      ; ILLEGAL DTE ADDRESS
```

```
**FILE**ID**VFGPVC




VV      VV   FFFFFFFFFF     GGGGGGGG   PPPPPPPP     VV      VV     CCCCCCCC
VV      VV   FFFFFFFFFF     GGGGGGGG   PPPPPPPP     VV      VV     CCCCCCCC
VV      VV   FF           GG           PP      PP   VV      VV   CC
VV      VV   FF           GG           PP      PP   VV      VV   CC
VV      VV   FF           GG           PP      PP   VV      VV   CC
VV      VV   FF           GG           PP      PP   VV      VV   CC
VV      VV   FFFFFFFF     GG           PPPPPPPP     VV      VV   CC
VV      VV   FFFFFFFF     GG           PPPPPPPP     VV      VV   CC
VV      VV   FF           GG  GGGGGG   PP           VV      VV   CC
VV      VV   FF           GG  GGGGGG   PP           VV      VV   CC
  VV  VV     FF           GG      GG   PP             VV  VV     CC               ....
  VV  VV     FF           GG      GG   PP             VV  VV     CC               ....
    VV       FF             GGGGGG     PP               VV         CCCCCCCC       ....
    VV       FF             GGGGGG     PP               VV         CCCCCCCC       ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                   .SBTTL  ALLXCB - ALLOCATE X25 CIRCUIT BLOCK

                                         ;+
                                         ; ALLXCB - ALLOCATE AND SET UP X25 CIRCUIT BLOCK
                                         ;
                                         ; INPUTS:
                                         ;          PRTADD - ADDRESS OF ENTRY IN PORT TABLE
                                         ;          $PFLAG - PSI CHARACTERISTICS WORD                                   ;[TD001]
                                         ;
                                         ; OUTPUTS:
                                         ;          R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK (APR 5 MAPPED TO BLOCK)
                                         ;          R0,R1,R2 DESTROYED
                                         ;
                                         ;-
000434  012701  000106                   ALLXCB: MOV     #X$LEN,R1          ; GET SIZE OF X.25 CIRCUIT BLOCK
000440  032767  000000G 000000G                  BIT     #PF.PSV,$PFLAG     ; CHECK FOR PSV MODULE                     ;[TD001]
000446  001402                                   BEQ     5$                 ; BRANCH IF NOT PRESENT                    ;[TD001]
000450  012701  000106                           MOV     #X$GLEN,R1         ; USE LONGER CIRCUIT BLOCK FOR GATEWAY     ;[TD001]
000454                                   5$:     CALL    $ALPOL             ; ALLOCATE CIRCUIT BLOCK                   ;[TD001]
000460  103500                                   BCS     20$                ; BR IF UNSUCCESSFUL ALLOCATION            ;**-1
000462  016702  000000G                          MOV     $PSIPT,R2          ; POINT TO PSI HOME BLOCK
000466  062702  000020                           ADD     #H$PTB,R2          ; POINT TO PORT TABLE LISTHEAD
000472                                           GETRV   R2,#2              ; READ IN PORT TABLE LISTHEAD
000510                                           SAVMAP                     ; SAVE CURRENT MAPPING
000514  016701  000004'                          MOV     PORTNO,R1          ; GET NUMBER OF PORTS
000520  006301                                   ASL     R1                 ; GET NUMBER OF BYTES IN PORT TABLE
000522  016746  000000G                          MOV     $BFR,-(SP)         ; SAVE UNMAPPED PORT TABLE ADDRESS
000526                                           CALL    $CEACC             ; MAP TO PORT TABLE
000532                                           GETAD   (SP)+,R1           ; READ IN PORT TABLE
000546  010077  000006'                          MOV     R0,@PRTADD         ; STORE UNMAPPED ADDRESS IN PORT TABLE
000552                                           PUTAD                      ; WRITE OUT UPDATED PORT TABLE
000556                                           RESMAP                     ; RESTORE PREVIOUS MAPPING
000562                                           CALL    INIXCB             ; INITIALIZE X25 CIRCUIT BLOCK
000566                                           CALL    FNDHSH             ; FIND HASH TABLE ENTRY AND SET UP
000572  103433                                   BCS     20$                ; BR IF NO ENTRY AVAILABLE
000574  012700  000106                           MOV     #X$LEN,R0          ; GET LENGTH OF XCB                        ;[TD001]
000600  032767  000000G 000000G                  BIT     #PF.PSV,$PFLAG     ; CHECK FOR PSV MODULE                     ;[TD001]
000606  001402                                   BEQ     8$                 ; BRANCH IF NOT PRESENT                    ;[TD001]
000610  012700  000106                           MOV     #X$GLEN,R0         ; PSV USES LONGER XCB                      ;[TD001]
000614                                   8$:     GETAD   R3,R0              ; READ IN CIRCUIT BLOCK                    ;[TD001]
000630  016700  000020'                          MOV     PVCOWN,R0          ; GET PROCESS OWNER NAME                   ;**-1
000634  001406                                   BEQ     10$                ; BR IF NONE SPECIFIED
000636  016701  000030'                          MOV     DLMSLN,R1          ; GET DLM'S SYSTEM LINE NUMBER
000642  052701  000200                           BIS     #200,R1            ; INDICATE SLN IS PRESENT
000646  110167  000023G                          MOVB    R1,X$MOWN+$BFR     ; STORE OWNER
000652  005067  000030G                  10$:    CLR     X$TCLZ+$BFR        ; STORE TIME SINCE LAST ZEROED
000656                                           PUTAD                      ; WRITE OUT UPDATED CIRCUIT BLOCK
000662                                   20$:    RETURN
```

```
VFGPVC     CREATED BY  MACRO  ON 29-JUN-85 AT 02:30      PAGE 3

SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

N$$VCT   = ******            15-498     16-536     17-586
PCVKW      000000 RG          9-193    #10-212
PF.DLM   = 000001            #7-145
PF.PSV   = ******    GX      14-439     14-460     16-538
PORTNO     000004 R          #7-133     14-448    *15-495     15-499     15-512
PRTADD     000006 R          #7-134     14-453    *15-509
PVCCTM     000012 R          #7-136    *11-339     16-550
PVCDF      000000 R           9-191
PVCFLG     000035 R          #7-147    *11-311    *12-393
PVCLCN     000010 R          #7-135    *11-327     16-553     17-590     17-607
PVCNAM     000037 R          #7-149     19-646
PVCNML     000036 R          #7-148     11-304
PVCOWN     000020 R          #7-139    *11-310    *12-379     14-464
PVCPRT     000034 R          #7-146    *15-510     16-552     17-608
PVCPSZ     000014 R          #7-137    *11-308    *12-352     16-549
PVCST      000000 RG        #10-212
PVCT       000166 R         #11-337
PVCTL      001554 R         #19-665
PVCWSZ     000016 R          #7-138    *11-309    *12-364     16-548
PVDLM      001506 R         #19-646
PVEND      000346 R         #13-404
PVFG       000314 R         #12-389
PVLCN      000130 R         #11-321
PVNAM      000056 R         #11-303
PVNAMX   = 000006            #6-105     11-305
PVOWN      000256 R         #12-374
PVPSZ      000212 R         #12-350
PVSTA      001626 R         #19-677
PVWSZ      000234 R         #12-362
PX$DLM   = 000200            20-746
RTSPC    = ******    GX       8-165      8-166      8-167      8-168      8-169      8-170      8-171      8-172
R$R0     = 000002            #6-111
R$R1     = 000004            #6-112
R$R2     = 000006            #6-113
R$R3     = 000010            #6-114
R$R4     = 000012            #6-115
R$R5     = 000014            #6-116
SF.ACT   = 000200            #5-70
SF.ENA   = 000100            #5-70
SF.LPB   = 000004            #5-70
SF.MFL   = 000040            #5-70
SF.PAC   = 000020            #5-70
SF.REA   = 000010            #5-70
SF.SER   = 000001            #5-70
SF.SVC   = 000002            #5-70
SF.UNL   = 000040            #5-70
SPACE    = 000040            #6-104
SP$P4      000062 RG         #7-158     16-562
STORID   = ******    GX      11-306
SYNERR   = ******    GX      *9-196     *9-199    *19-673
SYSFDB   = ******    GX     *14-446    *14-452    *14-463    *15-494    *15-497    *15-501    *16-541    *17-587    *17-598
                            *20-731    *20-736
```

```
    240
    241 000052                                  STATE$
    242 000052                                   TRAN$   <',>
    243
    244 000052                                  STATE$                  ; START OF DTE ADDRESS
    245 000052                                   TRAN$   <',>,SVOWN,NODTE ; NO DTE ADDRESS SPECIFIED
    246 000052                                   TRAN$   $DIGIT,,SVDTES
    247
    248 000052                                  STATE$  SVCDTE
    249 000052                                   TRAN$   $DIGIT,SVCDTE,SVDTED ; NEXT DIGIT IN DTE ADDRESS
    250 000052                                   TRAN$   <',>,,SVDTEE         ; END OF DTE ADDRESS
    251
    252 000052                                  STATE$  SVOWN           ; OWNER
    253 000052                                   TRAN$   $RAD50,$EXIT,SVOWNR
    254
    255 000052                                  STATE$
```

```
   3576 016472                                 SAVRG   <R0>                        ; SAVE R0
   3577 016474                                 CEACC$  R0                          ; CONVERT TO MAPPED ADDRESS
   3578 016504  016767  000002G 000000G        MOV     F.CLS+$BFR,$RSIZE           ; GET LENGTH OF BLOCK TO DEALLOCATE
   3579 016512  042767  177700  000000G        BIC     #^C<FF.MSK>,$RSIZE          ; ...
   3580 016520                                 PUTAD   R0                          ; WRITE OUT UPDATED FILTER BLOCK
   3581 016530                                 RESRG   <R0>                        ; RESTORE UNMAPPED ADDRESS
   3582 016532  000241                         CLC                                 ; FILTER BLOCK STILL USEFUL
   3583 016534  000555                         BR      80$                         ; FINISH UP
   3584
   3585 016536                         40$:    SAVRG   <R0>                        ; SAVE UNMAPPED ADDRESS
   3586 016540                                 CEACC$  R0                          ; CONVERT TO MAPPED ADDRESS
   3587 016550  016767  000002G 000000G        MOV     F.CLS+$BFR,$RSIZE           ; GET LENGTH OF BLOCK TO DEALLOCATE
   3588 016556  042767  177700  000000G        BIC     #^C<FF.MSK>,$RSIZE          ; ...
   3589 016564                                 PUTAD   R0                          ; WRITE OUT UPDATED FILTER BLOCK
   3590 016574                                 RESRG   <R0>                        ; RESTORE UNMAPPED ADDRESS
   3591 016576  012705  000014G                MOV     #$BFR+F.SEV+10,R5           ; POINT TO END OF 'SET' EVENT MASKS
   3592 016602                                 CALL    CHKEVT                      ; ANY MASKS SET?
   3593 016606  103130                         BCC     80$                         ; BR IF YES
   3594 016610  012705  000040G                MOV     #$BFR+F.CEV+10,R5           ; POINT TO END OF 'CLEAR' MASKS
   3595 016614                                 CALL    CHKEVT                      ; CHECK IF ANY MASKS SET
   3596 016620  000523                         BR      80$                         ; FINISH UP
   3597                                ;
   3598                                ; OPERATE ON GLOBAL FILTER BLOCK
   3599                                ;
   3600 016622  016767  000000G 000170' 60$:    MOV     $BFR,FLTBUF                 ; STORE POINTER TO NEXT BLOCK
   3601 016630  012703  000004G                MOV     #LO.EVT+RQB,R3              ; POINT TO FILTER MASKS
   3602 016634  012705  000004G                MOV     #$BFR+F.EVT,R5              ; POINT TO FILTER MASKS
   3603 016640                                 CALL    CLREVT                      ; CLEAR THE FILTER MASKS
   3604 016644  103013                         BCC     65$                         ; BR IF NOT NECESSARY TO DEALLOCATE
   3605 016646  012703  000004G                MOV     #$BFR+F.EVT,R3              ; POINT TO GLOBAL FILTER MASKS
   3606 016652  012705  000076'                MOV     #TMPMSK,R5                  ; POINT TO TEMPORARY STORAGE
   3607 016656                                 CALL    SAVCOM                      ; SAVE COMPLEMENT OF FILTER MASKS
   3608 016662  012705  000000G                MOV     #$BFR,R5                    ; DEALLOCATE FILTER BLOCK FROM $BFR
   3609 016666                                 CALL    DEAFIL                      ; DEALLOCATE FILTER BLOCK
   3610 016672  000424                         BR      70$                         ; CONTINUE
   3611 016674  012703  000004G         65$:    MOV     #$BFR+F.EVT,R3              ; POINT TO GLOBAL FILTER MASKS
   3612 016700  012705  000076'                MOV     #TMPMSK,R5                  ; POINT TO TEMPORARY STORAGE
   3613 016704                                 CALL    SAVCOM                      ; SAVE COMPLEMENT OF FILTER MASKS
   3614 016710                                 SAVRG   <R0>                        ; SAVE UNMAPPED ADDRESS
   3615 016712                                 CEACC$  R0                          ; CONVERT TO MAPPED ADDRESS
   3616 016722                                 PUTAD   R0,#F.LEN                   ; REWRITE GLOBAL FILTER BLOCK
   3617 016740                                 RESRG   <R0>                        ; RESTORE UNMAPPED ADDRESS
   3618 016742  010001                         MOV     R0,R1                       ; SAVE ADDRESS OF PREVIOUS BLOCK
   3619
   3620 016744                         70$:    SWTCHO  #FLTBUF                     ; USE FLTBUF FOR OUTPUT BUFFER
   3621 016752                         75$:    CALL    GETNXT                      ; GET NEXT FILTER BLOCK
   3622 016756  103044                         BCC     80$                         ; BR IF FINISHED
   3623 016760  012705  000220'                MOV     #FLTBUF+F.CEV,R5            ; POINT TO 'CLEAR' FILTER MASKS
   3624 016764  012703  000076'                MOV     #TMPMSK,R3                  ; POINT TO SCRATCH AREA
   3625 016770                                 CALL    CLREVT                      ; CLEAR THE 'CLEAR' MASKS NOT IN GLOBAL
   3626 016774  103013                         BCC     77$                         ; BR IF SOME MASKS STILL SET
   3627 016776  012705  000204'                MOV     #FLTBUF+F.SEV+10,R5         ; POINT TO END OF 'SET' FILTER MASKS
   3628 017002                                 CALL    CHKEVT                      ; ANY MASKS SET?
   3629 017006  103006                         BCC     77$                         ; BR IF YES
   3630 017010  012705  000170'                MOV     #FLTBUF,R5                  ; DEALLOCATE FILTER BLOCK FROM FLTBUF
   3631 017014  010400                         MOV     R4,R0                       ; GET ADDRESS OF FILTER BLOCK
   3632 017016                                 CALL    DEAFIL                      ; ELSE, DEALLOCATE FILTER BLOCK
```

```
                                         .SBTTL  STQUAL - SET QUALIFIER BITS

                                 ;+
                                 ;
                                 ; STQUAL - SET QUALIFIER BITS FOR FILTER BLOCK
                                 ;
                                 ; INPUTS:
                                 ;       $OPTON - OPTIONS WORD
                                 ;       $QUAL  - COMMAND QUALIFIER
                                 ;
                                 ; OUTPUTS:
                                 ;       EVQUAL - CONTAINS QUALIFIER BITS
                                 ;
                                 ;-

017650  005067  000120'          STQUAL: CLR     EVQUAL                  ; CLEAR QUALIFIER FLAGS WORD
017654  032767  000004  000000G          BIT     #QF$KNO,$QUAL           ; IS THIS FOR KNOWN LOGGING?
017662  001021                           BNE     10$                     ; BR IF YES
017664  012767  000002  000120'          MOV     #FF.FIL,EVQUAL          ; ASSUME LOGGING FILE
017672  032767  000200  000000G          BIT     #OP$FIL,$OPTON          ; IS THIS FOR LOGGING FILE?
017700  001012                           BNE     10$                     ; BR IF YES
017702  012767  000001  000120'          MOV     #FF.CON,EVQUAL          ; ASSUME LOGGING CONSOLE
017710  032767  000400  000000G          BIT     #OP$CON,$OPTON          ; IS THIS FOR LOGGING CONSOLE?
017716  001003                           BNE     10$                     ; BR IF YES
017720  012767  000004  000120'          MOV     #FF.MON,EVQUAL          ; ELSE MUST BE LOGGING MONITOR
017726  032767  000040  000000G  10$:    BIT     #OP$LNE,$OPTON          ; WAS A LINE QUALIFIER SPECIFIED?
017734  001403                           BEQ     20$                     ; BR IF NO
017736  052767  000010  000120'          BIS     #FF.LIN,EVQUAL          ; ELSE SET QUALIFIER BIT
017744  032767  000020  000000G  20$:    BIT     #OP$NOD,$OPTON          ; WAS A NODE QUALIFIER SPECIFIED?
017752  001403                           BEQ     30$                     ; BR IF NO
017754  052767  000020  000120'          BIS     #FF.ADD,EVQUAL          ; ELSE SET QUALIFIER BIT
017762  032767  000010  000000G  30$:    BIT     #OP$SIN,$OPTON          ; WAS A SINK NODE SPECIFIED?
017770  001403                           BEQ     40$                     ; BR IF NO
017772  052767  100000  000120'          BIS     #FF.REM,EVQUAL          ; ELSE SET QUALIFIER BIT
020000  032767  000001  000000G  40$:    BIT     #OP$HOS,$OPTON          ; WAS $HOST SPECIFIED?
020006  001403                           BEQ     50$                     ; BR IF NO
020010  052767  040000  000120'          BIS     #FF.HST,EVQUAL          ; ELSE SET QUALIFIER BIT
020016                           50$:    RETURN



020020                                   RETURN
```

```
                                        .SBTTL  TSKSCH - SEARCH FOR TASK TCB
                                ;+
                                ; **-TSKSCH-SEARCH TCB'S FOR SPECIFIED TASK
                                ;
                                ; INPUTS:
                                ;       R0      - POINTER TO RAD50 TASK NAME
                                ;
                                ; OUTPUTS:
                                ;       $RADDR  - ADDRESS OF TCB
                                ;       $BFR    - IMAGE OF TCB
                                ;       C-BIT SUCCESS/FAILURE
                                ;       R3,R4,R5 DESTROYED
                                ;-

021630                          TSKSCH:
021630                                  GETRV   .TSKHD,.TLGTH     ; READ IN LISTHEAD
021650  012705  000000G                 MOV     #$BFR,R5          ; POINT AT BUFFER BASE
021654  011565  000030                  MOV     (R5),T.TCBL(R5)   ; INITIALIZE FOR LOOP

021660  016567  000030  000000G 10$:    MOV     T.TCBL(R5),$RADDR ; SET TCB ADDRESS
021666  000261                          SEC                       ; ASSUME ERROR
021670  001411                          BEQ     30$               ; NULL TASK ( END OF LIST )
021672                                  GETRV                     ; READ IN TCB
021676  021065  000006                  CMP     (R0),T.NAM(R5)    ; MUST MATCH BOTH
021702  001366                          BNE     10$
021704  026065  000002  000010          CMP     2(R0),T.NAM+2(R5) ; HALVES OF NAME
021712  001362                          BNE     10$               ; C-BIT IS CLEAR
021714                          30$:    RETURN
```

```
VENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 6
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE           REFERENCES

E.LIN     000024        #5-79
E.LNK     000000        #5-79
E.MOD     000036        #5-79
E.NOD     000034        #5-79
E.PDV     000021        #5-79
E.PORT    000040        #5-79
E.PRM     000026        #5-79
E.PVC     000044        #5-79
E.SIZ     000022        #5-79
E.TIME    000004        #5-79
FATAL   = ****** GX      11-514     11-516     39-2057    39-2062    39-2063    39-2064    39-2065    39-2069    40-2171
                         40-2176    40-2177    40-2178    40-2179    83-4396
FF.ADD  = 000020        #5-69       5-69       57-3172    61-3449    73-3974
FF.CIR  = 000040        #5-69       5-69
FF.CON  = 000001        #5-69       6-166      55-3089    55-3092    63-3689    63-3691    73-3965
FF.FIL  = 000002        #5-69       6-167      55-3089    55-3095    63-3689    63-3691    73-3962
FF.HST  = 040000        #5-69       71-3909    73-3980
FF.LIN  = 000010        #5-69       5-69       45-2601    57-3172    61-3453    73-3971
FF.MOD  = 000100        #5-69       5-69
FF.MON  = 000004        #5-69       6-168      55-3089    55-3098    63-3689    63-3691    73-3968
FF.MSK  = 000077        #5-69       45-2595    60-3381    61-3435    61-3440    62-3539    62-3579    62-3588    62-3637
                         63-3679    63-3685    72-3931
FF.PRT  = 000200        #5-69       5-69
FF.QUL  = 000370        #5-69
FF.REM  = 100000        #5-69       60-3385    60-3387    60-3391    61-3445    63-3694    63-3696    63-3699    73-3977
FILDEV    000146 R      #6-154      50-2816   *50-2819    50-2826
FILLEN  = 000020        #6-128      50-2806    58-3284
FILNAM    000134 R      #6-151      47-2691    50-2805    50-2862    58-3282
FILOPR    015772 R       57-3220   #62-3481
FILPRO    014422 R       56-3147   #57-3170
FILTYP    000142 R      #6-152     *50-2900   *50-2902
FILUIC    000152 R      #6-156     *50-2849   *50-2853
FILUNT    000150 R      #6-155     *50-2820
FILVER    000144 R      #6-153     *50-2915
FINFLT    015340 R       59-3345   #60-3371
FINOBJ    005376 R      #29-1611    41-2276
FINREM    005500 R       28-1578   #30-1640
FLTBUF    000170 R      #6-161      62-3520   *62-3521    62-3526    62-3529    62-3532   *62-3600    62-3620    62-3623
                         62-3627    62-3630    63-3672    63-3673    63-3684    63-3688    63-3696    63-3699    63-3701
FNDCTR    021022 R       79-4205    80-4225   #81-4251
FNDDCB    020616 R       38-1909    50-2827    52-2995   #77-4159
FNDDMP    002732 R       13-604    #22-1015
FNDFIL    015534 R       57-3173    57-3183   #61-3425
FNDFR     002136 R       16-701    #18-820
FNDLIN    002304 R       15-669    #20-899
FNDPCL    002526 R       13-599    #21-961
FNDTRI    002310 R       13-594     14-637    #20-902
FNPID     002222 R       16-697    #19-857
FR$BCC  = 000007        #5-79
FR$CCF  = 000001        #5-79
FR$CDF  = 000002        #5-79
FR$DAO  = 000011        #5-79
```

```
VENA      CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 19

SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE           REFERENCES

$QUAL    = ******   GX    41-2203    41-2210    41-2223    42-2307    43-2465    45-2561    47-2689    55-3090    56-3126
                          56-3150    58-3244    73-3960
$RADDR   = ******   GX   *11-442    *20-919    *20-927    *21-977    *21-984    *22-1031   *22-1037   *26-1267   *26-1290
                         *26-1307   *26-1324   *26-1341   *26-1358   *27-1399   *27-1407   *28-1570   *28-1573   *28-1583
                         *29-1620   *30-1649   *32-1713   *36-1830   *36-1832   *37-1870   *37-1872   *38-1911    39-1979
                         *39-1990   *39-1993   *39-2015   *39-2040   *39-2044   *39-2046    40-2124   *40-2133   *40-2136
                         *40-2157   *41-2273   *42-2339   *42-2345   *42-2390   *42-2423   *42-2427   *45-2593   *49-2774
                         *50-2831   *53-3020   *59-3341   *60-3379   *61-3433   *62-3503   *62-3515   *62-3540   *62-3580
                         *62-3589   *62-3616   *62-3638   *63-3677   *74-4009   *74-4025   *74-4028   *74-4065   *75-4094
                         *75-4103   *75-4123   *76-4139   *76-4142   *77-4161   *78-4179   *81-4271   *81-4275   *82-4321
                         *82-4328   *82-4333   *82-4344   *82-4348   *84-4415   *84-4419
$RBLCK   = ******   GX   *20-913    *20-918    *21-971    *21-976    *22-1025   *22-1030   *39-2038   *39-2045   *81-4272
$RDBLH   = ******   GX    39-2033    39-2034
$RPASW     006130 RG      25-1229    25-1247   #36-1827
$RSIZE   = ******   GX   *45-2594   *45-2595    45-2597   *60-3380   *60-3381    60-3382   *61-3434   *61-3435    61-3436
                         *62-3538   *62-3539   *62-3578   *62-3579   *62-3587   *62-3588   *62-3636   *62-3637   *63-3678
                         *63-3679    63-3680
$SAVAL   = ******   GX    11-397     45-2582
$SAVRG   = ******   GX    57-3170
$SLN     = ******   GX    11-416     11-436     16-715     16-717     16-735     16-738     45-2605    61-3455    71-3899
                          81-4268
$SLTA    = ******   GX    11-447     11-467     12-554     12-567     13-591     14-633     15-665     16-703     17-772
                          20-904     21-962     22-1016    24-1113    24-1123    24-1137    24-1147    81-4251
$SLTMA   = ******   GX    35-1807
$SMOD      007524 RG     #40-2084
$SNOD      003460 RG     #25-1171
$TPASW     006260 RG      25-1230    25-1248   #37-1867
$TRIB    = ******   GX    11-422     11-461     16-726     16-728     16-736     16-743     17-790     17-792     20-928
                          21-985     22-1038    81-4259    81-4269
$VACP      006524 RG      25-1232   #39-1939
$VACT      001350 RG     #14-632
$VALI      011012 RG     #42-2300
$VCOST     002006 RG     #17-771
$VDEA      001430 RG     #15-664
$VENA      000250 RG     #11-394
$VERIF     006410 RG      25-1231   #38-1907
$VFLAG   = ******   GX   *11-414     11-474     11-479     11-494    *11-510     11-518     12-539     23-1065    24-1098
                          26-1264    28-1490    28-1504    39-1940    39-1947    39-1984    39-2010   *39-2024    39-2031
                          40-2090    40-2108    40-2115    40-2129    40-2152    42-2365    53-3025    54-3061    55-3100
                          75-4120    83-4387    83-4394
$VHST      020662 RG      25-1226    25-1244   #78-4178
$VHTM      020770 RG     #80-4225
$VINA      004144 RG      25-1237   #26-1341
$VINC      004054 RG      25-1235   #26-1307
$VLIN      004234 RG      25-1227    25-1245   #27-1379
$VLOG      012002 RG     #44-2489
$VLTM      020736 RG     #79-4205
$VOBJ      010306 RG     #41-2196
$VOUT      004110 RG      25-1236   #26-1324
$VOWN      001510 RG     #16-695
$VREM      004654 RG      25-1228    25-1246   #28-1484
$VRET      004200 RG      25-1238   #26-1358
```

Table of contents

```
    14-  785    X3CH
    15-  843    X2CH
```

```
                                                  ;
                                                  ; SECONDARY FILE
                                                  ;
 001204   012767   100000   000450'  MXTAB:  MOV     #100000,FLAGS     ; INDICATE MUX EXPANSION
 001212   112546                     FILE:   MOVB    (R5)+,-(SP)       ; GET REPEAT COUNT
 001214   010546                             MOV     R5,-(SP)          ; SAVE CURRENT BINARY FILE POINTER
 001216   011605                     10$:    MOV     (SP),R5           ; RESTORE BINARY FILE POINTER
 001220   005003                     20$:    CLR     R3                ; GET FUNCTION CODE
 001222   152503                             BISB    (R5)+,R3          ;       ...
 001224   122703   000000G                   CMPB    #.EOF,R3          ; IS IT END OF FILE?
 001230   001406                             BEQ     30$               ; IF EQ, YES
 001232   106303                             ASLB    R3                ; CONVERT TO A WORD INDEX
 001234                                      CALL    @TABLE-2(R3)      ; EXPAND A FIELD
 001240                                      CALL    $BUFCK            ; DON'T LET BUFFER OVERFILL
 001244   000765                             BR      20$               ; KEEP LOOKING FOR EOF
 001246   005267   000446'           30$:    INC     INDEX             ; TALLY ONE PASS THRU SECONDARY FILE
 001252   126667   000002   000446'          CMPB    2(SP),INDEX       ; ARE WE DONE?
 001260   001356                             BNE     10$               ; IF NE, NO
 001262   022626                             CMP     (SP)+,(SP)+       ; PURGE STACK
 001264   005067   000446'                   CLR     INDEX             ; CLEAR SECONDARY INDEX
 001270   005067   000450'                   CLR     FLAGS             ; AND SPECIAL FLAGS
 001274                                      RETURN

                                                  ;
                                                  ; LINKED SECONDARY FILE
                                                  ;
 001276   012767   000200   000450'  MPTAB:  MOV     #200,FLAGS        ; INDICATE MTP EXPANSION
 001304   112546                     LFILE:  MOVB    (R5)+,-(SP)       ; SAVE REPEAT COUNT
 001306   112546                             MOVB    (R5)+,-(SP)       ; AND LISTHEAD DISPLACEMENT
 001310   112566   000001                    MOVB    (R5)+,1(SP)       ;       ...
 001314   010546                             MOV     R5,-(SP)          ; SAVE BINARY POINTER
 001316   011605                     10$:    MOV     (SP),R5           ; RESET BINARY POINTER
 001320   010446                             MOV     R4,-(SP)          ; SAVE SECONDARY TABLE ADDRESS
 001322   005003                     20$:    CLR     R3                ; GET FUNCTION CODE
 001324   152503                             BISB    (R5)+,R3          ;       ...
 001326   022703   000000G                   CMP     #.EOF,R3          ; IS IT END OF FILE?
 001332   001404                             BEQ     30$               ; IF EQ, YES
 001334   106303                             ASLB    R3                ; CONVERT TO WORD INDEX
 001336                                      CALL    @TABLE-2(R3)      ; EXPAND A FIELD
 001342   000767                             BR      20$               ; KEEP LOOKING FOR EOF
 001344   012700   000000G           30$:    MOV     #$TMPBF,R0        ; GET MULTIPOINT LISTHEAD ADDRESS
 001350   066600   000004                    ADD     4(SP),R0          ;       ...
 001354   012601                             MOV     (SP)+,R1          ; RESTORE SECONDARY TABLE ADDRESS
 001356   016002   000002                    MOV     2(R0),R2          ; LINK SECONDARY TABLE TO LISTHEAD
 001362   166702   000000G                   SUB     $LINK,R2          ; RELOCATE TO SCOM
 001366   005011                             CLR     (R1)              ;       ...
 001370   066701   000000G                   ADD     $LINK,R1          ; RELOCATE TO SCOM
 001374   010160   000002                    MOV     R1,2(R0)          ;       ...
 001400   010112                             MOV     R1,(R2)           ;       ...
 001402   005267   000446'                   INC     INDEX             ; TALLY ONE PASS THRU SECONDARY FILE
 001406   126667   000004   000446'          CMPB    4(SP),INDEX       ; ARE WE DONE?
 001414   001340                             BNE     10$               ; IF NE, NO
 001416   062706   000006                    ADD     #6,SP             ; PURGE STACK
 001422   005067   000446'                   CLR     INDEX             ; CLEAR SECONDARY INDEX
 001426   005067   000450'                   CLR     FLAGS             ; AND SPECIAL FLAGS
 001432                                      RETURN
```

```
VEX2       CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 4
SYMBOL CROSS REFERENCE                                   CREF   04.00
SYMBOL   VALUE            REFERENCES

$ICBAD   = ******   GX      12-670
$ICBLN   = ******   GX      12-659
$ICBSZ   = ******   GX      12-670
$LDA     = ******   GX      12-590     *12-593
$LINK    = ******   GX      7-271       7-316       11-560      11-562      13-714     *13-715     *13-716     *13-717     *13-760
$MXADD   = ******   GX      13-715
$MXSIZ   = ******   GX      13-759
$PTR     = ******   GX      6-181
$PUTRC   = ******   GX      12-670      13-748
$RADDR   = ******   GX     *7-306      *7-373      *12-670     *13-748
$SADDR   = ******   GX      7-324       12-602      12-652
$SLN     = ******   GX      7-352       7-364
$SLTA    = ******   GX      7-269
$TMPBF   = ******   GX      11-556
$UDA     = ******   GX      8-458      *8-470       9-483      *9-486
$VECT1   = ******   GX      12-653     *12-658
$VEX2      000000 RG       #6-175
$X2HLD   = ******   GX      15-867
$X2MBS   = ******   GX      15-852
$X2MRT   = ******   GX      15-857
$X2MWS   = ******   GX      15-847
$X2RET   = ******   GX      15-862
$X3CAL   = ******   GX      14-810
$X3CLR   = ******   GX      14-815
$X3DBS   = ******   GX      14-790
$X3DWS   = ******   GX      14-800
$X3MBS   = ******   GX      14-795
$X3MCL   = ******   GX      14-830
$X3MRE   = ******   GX      14-835
$X3MRS   = ******   GX      14-840
$X3MWS   = ******   GX      14-805
$X3RES   = ******   GX      14-820
$X3RST   = ******   GX      14-825
$$APR    = ******   GX      13-773
$$CSR    = ******   GX      7-244
$$DCHA   = ******   GX      7-274       7-275
$$DPR    = ******   GX      13-779      13-782
$$LSAD   = ******   GX      7-202       7-211
$$MTP    = ******   GX      7-391       7-404
$$PCHA   = ******   GX      7-294
$$PRI    = ******   GX      10-513
$$SCSR   = ******   GX      7-339
$$SNUM   = ******   GX      7-394       7-408
.EOF     = ******   GX      11-525      11-551      13-727
.MGMGE   = ******   GX      8-447       8-460       10-494      12-586      12-600
```

```
VEX3      CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00
SYMBOL  VALUE              REFERENCES
NULL      000054 R          5-77     5-78     5-84     5-85     5-86     5-90     5-92     5-93     5-96
                            5-97     5-100    5-101    5-102    5-103    5-104    5-105    5-106    5-113
                            5-114    5-115    5-116    5-117    5-118    5-119    5-120    5-121    5-122
                            5-123    5-124    5-125    5-126    5-127    5-128    5-129   #7-186
POP1      000050 R          5-79     5-80     5-81     5-83     5-87     5-95     5-98     5-108    5-109
                           #7-180
POP2      000044 R          5-88     5-107    5-110    5-111   #7-174
POP3      000036 R          5-91     5-94    #7-168
TABLE     000000 R         #5-77     6-156
XCORE     000106 R          5-82    #8-208
XLIBR     000146 R          5-89    #9-227
XMXPTB    000202 R          5-112  #10-242
XSCOM     000056 R          5-99    #8-195
$ADDR   = ******  GX        6-149
$BIN    = ******  GX        6-148
$DEA16  = ******  GX        8-202    10-253
$DEA18  = ******  GX        8-221
$MXLEN  = ******  GX        10-250
$PTR    = ******  GX        6-150
$VEX3     000000 RG        #6-147
$XLBR   = ******  GX        9-237
.MGMGE  = ******  GX        8-212    9-234
```

```
                                         ;
                                         ; SECONDARY FILE
                                         ;
001044  005267  000154'                  MXTAB:  INC     FLAGS              ; INDICATE MUX EXPANSION
001050  112546                           FILE:   MOVB    (R5)+,-(SP)        ; GET REPEAT COUNT
001052  010546                                   MOV     R5,-(SP)           ; SAVE CURRENT BINARY POINTER
001054  011605                           10$:    MOV     (SP),R5            ; RESTORE BINARY POINTER
001056  005767  000154'                          TST     FLAGS              ; IS IT MUX EXPANSION ?
001062  001407                                   BEQ     20$                ; NO ..
001064  126467  000013  000152'                  CMPB    L.UNT(R4),INDEX    ; DO UNIT NUMBERS MATCH ?
001072  001003                                   BNE     20$                ; NO ..
001074  026403  000004                           CMP     L.DDS(R4),R3       ; IS THE LINE TABLE POINTER CORRECT ?
001100  001025                                   BNE     101$               ; NO .. ERROR
001102  005002                           20$:    CLR     R2                 ; GET FUNCTION CODE
001104  152502                                   BISB    (R5)+,R2           ; ...
001106  122702  000000G                          CMPB    #.EOF,R2           ; IS IT END OF FILE?
001112  001404                                   BEQ     30$                ; IF EQ, YES
001114  106302                                   ASLB    R2                 ; CONVERT TO A WORD INDEX
001116                                           CALL    @TABLE-2(R2)       ; EXPAND A FIELD
001122  000767                                   BR      20$                ; KEEP LOOKING FOR EOF
001124  005267  000152'                  30$:    INC     INDEX              ; TALLY ONE PASS THRU SECONDARY FILE
001130  126667  000002  000152'                  CMPB    2(SP),INDEX        ; ARE WE DONE?
001136  001346                                   BNE     10$                ; IF NE, NO
001140  022626                                   CMP     (SP)+,(SP)+        ; PURGE STACK
001142  005067  000152'                          CLR     INDEX              ; CLEAR SECONDARY INDEX
001146  005067  000154'                          CLR     FLAGS              ; AND SPECIAL FLAGS
001152                                           RETURN

                                         ;
                                         ; ERROR CONDITION
                                         ;
001154                                   101$:   CRASH                      ; LINE TABLE POINTER (R3) IS SCREWED UP

                                         ;
                                         ; LINKED SECONDARY FILE
                                         ;
001156                                   MPTAB:  CRASH
001160  112546                           LFILE:  MOVB    (R5)+,-(SP)        ; SAVE REPEAT COUNT
001162  122525                                   CMPB    (R5)+,(R5)+        ; DISCARD LISTHEAD DISPLACEMENT
001164  010546                                   MOV     R5,-(SP)           ; SAVE CURRENT BINARY POINTER
001166  011605                           10$:    MOV     (SP),R5            ; RESET BINARY POINTER
001170  005002                           20$:    CLR     R2                 ; GET FUNCTION CODE
001172  152502                                   BISB    (R5)+,R2           ; ...
001174  022702  000000G                          CMP     #.EOF,R2           ; IS IT END OF FILE?
001200  001404                                   BEQ     30$                ; IF EQ, YES
001202  106302                                   ASLB    R2                 ; CONVERT TO WORD INDEX
001204                                           CALL    @TABLE-2(R2)       ; EXPAND A FIELD
001210  000767                                   BR      20$                ; KEEP LOOKING FOR EOF
001212  005267  000152'                  30$:    INC     INDEX              ; TALLY ONE PASS THRU SECONDARY FILE
001216  126667  000002  000152'                  CMPB    2(SP),INDEX        ; ARE WE DONE?
001224  001360                                   BNE     10$                ; IF NE, NO
001226  022626                                   CMP     (SP)+,(SP)+        ; PURGE STACK
001230  005067  000152'                          CLR     INDEX              ; CLEAR SECONDARY INDEX
001234  005067  000154'                          CLR     FLAGS              ; AND SPECIAL FLAGS
001240                                           RETURN
```

Table of contents

```
     5-   55    MACRO DEFINITIONS
     6-   79    LOCAL SYMBOL DEFINITIONS
     7-   99    LOCAL DATA
     8-  126    ERROR MESSAGES
     9-  138    LOOK FOR DTE$DF MACRO
    10-  169    TPARS STATE TABLES
    11-  221    LINE-ID SUBEXPRESSION
    12-  244    DTE$DF ACTION ROUTINES
    16-  361    CHKDTE - CHECK FOR VALID DTE ADDRESS
    17-  403    BIASX - SET APR6 BIAS CORRECTLY
    18-  424    LINE-ID ACTION ROUTINES
    19-  455    STRNXT - STORE NEXT DIGIT
    20-  481    PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
```

```
                                                .SBTTL  CHKDTE - CHECK FOR VALID DTE ADDRESS

                                        ;+
                                        ;
                                        ; CHKDTE - CHECK FOR VALID DTE ADDRESS
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       CARRY CLEAR - DTE ADDRESS IS UNIQUE
                                        ;               R4 = COUNT OF DTES DEFINED ALREADY
                                        ;       CARRY SET - DTE ADDRESS IS NOT UNIQUE
                                        ;       R0,R1,R2 DESTROYED
                                        ;
                                        ;-
000426                                  CHKDTE: CALL    $SAVAL                  ; SAVE ALL REGISTERS
000432  016700  000000G                         MOV     $PSIPT,R0               ; POINT TO PSI HOME BLOCK
000436  062700  000002                          ADD     #H$LDTE,R0              ; POINT TO LOCAL DTE LISTHEAD
000442                                          GETRV   R0,#2                   ; READ IN LISTHEAD
000460                                          SAVMAP                          ; SAVE CURRENT MAPPING
000464  005004                                  CLR     R4                      ; INITIALISE DTE COUNT
000466  016700  000000G                 10$:    MOV     $BFR,R0                 ; POINT TO NEXT BLOCK IN LIST
000472  000261                                  SEC                             ; INDICATE NOT IN DTE LIST??
000474  001433                                  BEQ     30$                     ; BR IF END OF LIST
000476  010067  000000G                         MOV     R0,DTEDES               ; SAVE UNMAPPED DTE ADDRESS
000502  005204                                  INC     R4                      ; ONE MORE BLOCK SEEN
000504                                          CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
000514                                          GETAD   R0,#L$LEN               ; READ IN DTE DESCRIPTOR
000532  012701  000010                          MOV     #<DTEALN+1>/2,R1        ; GET DTE ADDRESS LENGTH
000536  012702  000000G                         MOV     #$BFR,R2                ; GET ADDRESS OF LOCAL DTE DESCRIPTOR
000542  062702  000020                          ADD     #L$DTEA,R2              ; POINT TO DTE ADDRESS
000546  012703  000017'                         MOV     #DTEPCK,R3              ; POINT TO SPECIFIED DTE ADDRESS
000552  122223                          20$:    CMPB    (R2)+,(R3)+             ; DO THE ADDRESSES MATCH?
000554  001344                                  BNE     10$                     ; BR IF NO: UNIQUE SO FAR
000556  005301                                  DEC     R1                      ; MORE OF ADDRESS TO CHECK?
000560  003374                                  BGT     20$                     ; BR IF YES
000562  000241                                  CLC                             ; INDICATE FOUND IT
000564                                  30$:    RESMAP                          ; RESTORE PREVIOUS MAPPING
000570  010466  000012                          MOV     R4,R$R4(SP)             ; RETURN COUNT IN TASK R4
000574                                  40$:    RETURN
```

```
VFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:29
Table of contents
```

```
     5-   54     MACRO DEFINITIONS
     6-   99     LOCAL SYMBOL DEFINITIONS
     7-  118     LOCAL DATA
     8-  161     ERROR MESSAGES
     9-  178     LOOK FOR PVC$DF MACRO
    10-  208     TPARS STATE TABLES
    11-  297     PVC$DF ACTION ROUTINES
    14-  423     ALLXCB - ALLOCATE X25 CIRCUIT BLOCK
    15-  473     FNDPRT - FIND FREE PORT ENTRY
    16-  519     INIXCB - INITIALIZE X25 CIRCUIT BLOCK
    17-  567     FNDHSH - FIND HASH TABLE ENTRY
    18-  613     BIASX - BIAS XPOOL ADDRESS FOR APR6
    19-  633     PVDLM - CHECK FOR DLM PVC
    20-  687     DLMTLC - SET UP TRANSPORT LINE COUNTERS
```

```
                                        .SBTTL  FNDPRT - FIND FREE PORT ENTRY

                                ;+
                                ;
                                ; FNDPRT - FIND FREE PORT ENTRY
                                ;
                                ; INPUTS:
                                ;       PORTNO - TOTAL NUMBER OF PORTS
                                ;
                                ; OUTPUTS:
                                ;       CARRY CLEAR - FREE PORT FOUND
                                ;               R1 - PORT NUMBER ALLOCATED
                                ;               PVCPRT - PORT NUMBER
                                ;               PRTADD - MAPPED ADDRESS OF ENTRY IN PORT TABLE
                                ;       CARRY SET - FREE PORT NOT FOUND
                                ;
                                ;-
000664                          FNDPRT: SAVRG   <R0,R2>                 ; SAVE R0,R2
000670                                  SAVMAP                          ; SAVE CURRENT MAPPING
000674  016700  000000G                 MOV     $PSIPT,R0               ; GET ADDRESS OF PSI HOME BLOCK
000700  062700  000022                  ADD     #H$NPT,R0               ; POINT TO NUMBR OF PORTS
000704                                  GETRV   R0,#2                   ; READ IN NUMBER OF PORTS
000722  016767  000000G 000004'         MOV     $BFR,PORTNO             ; SAVE NUMBER OF PORTS
000730  062700  177776                  ADD     #H$PTB-H$NPT,R0         ; POINT TO PORT TABLE LISTHEAD
000734                                  GETRV   R0,#2                   ; READ IN LISTHEAD
000752                                  CEACC$  $BFR,R0                 ; GET MAPPED ADDRESS
000764  016701  000004'                 MOV     PORTNO,R1               ; GET NUMBER OF PORTS
000770  006301                          ASL     R1                      ; GET NUMBER OF BYTES IN PORT TABLE
000772                                  GETAD   R0,R1                   ; READ IN PORT TABLE
001006  012702  000000G                 MOV     #$BFR,R2                ; POINT TO PORT TABLE
001012  005001                          CLR     R1                      ; PORT NUMBER COUNTER
001014  005722                          TST     (R2)+                   ; SKIP FIRST ENTRY (NOT AVAILABLE)
001016  005201                  10$:    INC     R1                      ; INCREMENT PORT NUMBER
001020  005722                          TST     (R2)+                   ; IS THIS A FREE PORT?
001022  001006                          BNE     15$                     ; BR IF NO
001024  005742                          TST     -(R2)                   ; POINT TO PORT ENTRY
001026  010267  000006'                 MOV     R2,PRTADD               ; SAVE MAPPED ADDRESS OF ENTRY
001032  110167  000034'                 MOVB    R1,PVCPRT               ; SAVE PORT NUMBER ALLOCATED
001036  000404                          BR      20$                     ; AND EXIT
001040  020167  000004'         15$:    CMP     R1,PORTNO               ; END OF LIST?
001044  103764                          BLO     10$                     ; BR IF NO
001046  000261                          SEC                             ; INDICATE NUMBER OF PORTS EXCEEDED
001050                          20$:    RESMAP                          ; RESTORE PREVIOUS MAPPING
001054                                  RESRG   <R2,R0>                 ; RESTORE R0,R2
001060                                  RETURN                          ;
```

```
VFGPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:30      PAGE 4

SYMBOL  CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE           REFERENCES

S$$BAS   = ******         8-165    8-165    8-166    8-166    8-167    8-167    8-168    8-168    8-169
                          8-169    8-170    8-170    8-171    8-171    8-172    8-172
S.COST     000001        #5-70
S.FLG      000000        #5-70
S.LEN      000004        #5-70
S.NMST     000002        #5-70
S.OWNR     000003        #5-70
T$FLAG     000044        #5-66    *20-746
T$LIF      000013        #5-66
T$LIFL     000013        #5-66
T$LIFO     000013        #5-66
T$LIFS     000013        #5-66
T$LIN      000000        #5-66     20-737   20-739   *20-744
T$LIPS     000006        #5-66
T$LLD      000012        #5-66
T$LLDC     000045        #5-66
T$LLDL     000012        #5-66
T$LLDO     000012        #5-66
T$LLDS     000012        #5-66
T$LLEN     000046        #5-66     20-736   20-741
T$LOPR     000002        #5-66
T$LTCL     000024        #5-66
T$LTIM     000026        #5-66
T$LTPR     000014        #5-66
T$LTPS     000020        #5-66
T$NAPL     000004        #5-66
T$NFE      000000        #5-66
T$NLEN     000010        #5-66
T$NNUL     000002        #5-66
T$NOPL     000006        #5-66
T$NRNI     000042        #5-66
T$NRPL     000005        #5-66
T$NRUL     000007        #5-66
T$NVR      000001        #5-66
T$RPRI     000040        #5-66
T$SVC      000034        #5-66    *20-745
T$T5       000030        #5-66    *20-747
T$T6       000032        #5-66    *20-748
XT$PVC   = 000001         16-563
X$ABQ      000076        *16-560   16-561   *16-561
X$CTIM     000060        *16-550
X$DTE      000024        *16-551
X$GLEN     000106         14-441   14-462    16-540
X$LCN      000026        *16-553
X$LEN      000106         14-438   14-459    16-537
X$MOWN     000023        *14-468
X$PKSZ     000016        *16-549
X$PRT      000021        *16-552
X$RXQ      000072        *16-556   16-557   *16-557
X$ST       000005        *16-562
X$TCLZ     000030        *14-469
X$TXQ      000066        *16-554   16-555   *16-555
```

```
                                                 .SBTTL  SVC$DF ACTION ROUTINES

                                                 .ENABL  LSB
                                         ;
                                         ; SVNAM - PROCESS NAME
                                         ;
000052  022767  000003  000000G  SVNAM:  CMP     #3,.PSTCN             ; VALID NAME?
000060  001063                           BNE     101$                  ; BR IF NO
000062  016700  000000G                  MOV     .PSTPT,R0             ; POINT TO PROCESS NAME
000066  005001                           CLR     R1                    ;
000070                                   CALL    $CAT5                 ; CONVERT TO RAD50
000074  022701  015355                   CMP     #^RDLM,R1             ; IS IT DLM?
000100  001053                           BNE     101$                  ; BR IF NO - ERROR
000102  005067  000006'                  CLR     PKTSVC                ; INITIALIZE OPTIONAL PARAMETERS
000106  105067  000022'                  CLRB    WNDSVC                ; ...
000112                                   RETURN

                                         ;
                                         ; SVCTL - CONTROLLER NUMBER
                                         ;
000114  005767  000000G          SVCTL:  TST     .PNUMH                ; VALID LINE-ID?
000120  001043                           BNE     101$                  ; BR IF NO
000122  105767  000001G                  TSTB    .PNUMB+1              ; ...
000126  001040                           BNE     101$                  ; BR IF INVALID LINE-ID
000130  116767  000000G 000021'          MOVB    .PNUMB,CTLSVC         ; SAVE CONTROLLER NUMBER
000136  016700  000000G                  MOV     $SLTA,R0              ; GET SLT ADDRESS
000142  126760  000000G 000012           CMPB    .PNUMB,L.CTL(R0)      ; CORRECT CONTROLLER NUMBER?
000150  001404                           BEQ     10$                   ; BR IF YES
000152                                   REJ$                          ; ELSE REJECT TRANSITION
000162                           10$:    RETURN

                                         ;
                                         ; SVSTA - STATION NUMBER
                                         ;
000164  005767  000000G          SVSTA:  TST     .PNUMH                ; VALID LINE-ID?
000170  001017                           BNE     101$                  ; BR IF NO
000172  105767  000001G                  TSTB    .PNUMB+1              ; ...
000176  001014                           BNE     101$                  ; BR IF INVALID ID
000200  116767  000000G 000014'          MOVB    .PNUMB,STASVC         ; SAVE STATION NUMBER
000206  126767  000000G 000000G          CMPB    $TRIB,.PNUMB          ; CORRECT STATION NUMBER?
000214  001404                           BEQ     20$                   ; BR IF YES
000216                                   REJ$                          ; ELSE REJECT TRANSITION
000226                           20$:    RETURN
                                         ;
                                         ; ERROR
                                         ;
000230                           101$:   EMSG$R  21                    ; ILLEGAL LINE-ID
                                         .DSABL  LSB
```

```
017022  000753                          BR      75$                     ; CHECK NEXT BLOCK IN LIST
017024                          77$:    SAVRG   <R4>                    ; SAVE UNMAPPED ADDRESS
017026                                  CEACC$  R4                      ; CONVERT TO MAPPED ADDRESS
017036  016767  000002G 000000G         MOV     F.CLS+$BFR,$RSIZE       ; GET LENGTH OF BLOCK TO DEALLOCATE
017044  042767  177700  000000G         BIC     #^C<FF.MSK>,$RSIZE      ; ...
017052                                  PUTAD   R4                      ; OUTPUT UPDATED FILTER BLOCK
017062                                  RESRG   <R4>                    ; RESTORE UNMAPPED ADDRESS
017064  010401                          MOV     R4,R1                   ; SAVE ADDRESS OF PREVIOUS
017066  000731                          BR      75$                     ; GET NEXT BLOCK IN LIST

017070                          80$:    SWTCHO                          ; RESTORE $BFR AS OUTPUT BUFFER
017076  012600                          MOV     (SP)+,R0                ; RESTORE R0
017100                                  RETURN
```

```

                                        .SBTTL  ALPLB - ALLOCATE PLB DATA BASE
                                ;+
                                ; *** - ALPLB - ALLOCATE PLB DATA BASE
                                ;       Allocates PLB's and index vector.  Initializes the vector and
                                ;       clears the PLBs.  Note: Vector and PLBs are allocated contiguously.
                                ;
                                ; INPUTS:
                                ;       NONE.
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       No registers preserved
                                ;-

020022                          ALPLB:
020022                                  CALL    CHKXPT                  ; GET XPT'S DATA BASE ADDRESS
                                                                        ; Ret: R0=PDV addr, R1=XPT PDV idx,
                                                                        ;      R2=XPT db virt addr
020026  103536                          BCS     40$                     ; IF CS, ASSUME A PROTOCOL EMULATOR
020030  116000  000000G                 MOVB    Z.LLN(R0),R0            ; GET NUMBER OF XPT CHANNELS

020034                                  GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK

020054  010067  000030G                 MOV     R0,$BFR+D$NLN           ; SAVE IT
020060  010046                          MOV     R0,-(SP)                ; ...
020062                                  PUTRC                           ; WRITE BACK DECNET HOME BLOCK

020066  012701  000052                  MOV     #P$LEN,R1               ; GET SIZE OF EACH PLB
020072                                  CALL    $MUL                    ; CALCULATE SIZE OF PLB DATA BASE
020076  012600                          MOV     (SP)+,R0                ; Get number of channels again
020100  006300                          ASL     R0                      ; multiply by two for vector size
020102  060001                          ADD     R0,R1                   ; add to total allocation
020104                                  CALL    $ALL15                  ; TRY TO ALLOCATE PLB'S
020110  103506                          BCS     50$                     ; IF CS, FAILED TO ALLOCATE PLB'S

                                ; Link vector into XPT's data base

020112                                  GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
020132  016705  000030G                 MOV     $BFR+D$NLN,R5           ; SAVE NUMBER OF CIRCUITS

020136                                  GETRV   R2,#N$XLEN              ; READ IN XPT'S DATA BASE
020154  010567  000016G                 MOV     R5,N$PLD+$BFR           ; STORE NUMBER OF PLB ENTRIES
020160  010067  000020G                 MOV     R0,N$PLD+2+$BFR         ; AND ADDRESS OF PLB DATA BASE
020164                                  PUTRC                           ; WRITE BACK THE XPT DATA BASE

                                ;
                                ; Initialize Vector and PLBs
                                ;
020170  105067  000123'                 CLRB    LCLFLG                  ; Clear flag for first time
020174  010102                          MOV     R1,R2                   ; SAVE NUMBER OF BYTES ALLOCATED
020176  020227  001000          10$:    CMP     R2,#512.                ; CAN PUTRV HANDLE THIS TRANSFER?
020202  003403                          BLE     20$                     ; BR IF YES
020204  012701  001000                  MOV     #512.,R1                ; SET UP FOR LARGEST TRANSFER
020210  160102                          SUB     R1,R2                   ; UPDATE NUMBER OF BYTES REMAINING

020212  010046                  20$:    MOV     R0,-(SP)                ; Push Virtual address of block
020214  010146                          MOV     R1,-(SP)                ; and byte count of transfer
```

```
                                        .SBTTL  $FPDV - FIND A PDV ADDRESS
                                ;+
                                ; $FPDV - TRY TO FIND PDV
                                ;
                                ; INPUTS:
                                ;       R0 = RAD50 PDV NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0 = PDV ADDRESS
                                ;       R1 = PDV index
                                ;       R2 = DESTROYED
                                ;
                                ;-
021716                          $FPDV::
021716  010046                          MOV     R0,-(SP)          ; SAVE NAME TO BE MATCHED
021720  016701  000000G                 MOV     $PDVTA,R1         ; GET PDV TABLE ADDRESS
021724  016702  000000G                 MOV     $PDVNM,R2         ; AND NUMBER OF PDV'S
021730  001407                          BEQ     30$               ; IF ZERO, NO PDV'S!
021732  012100                  10$:    MOV     (R1)+,R0          ; GET PDV ADDRESS
021734  001403                          BEQ     20$               ; IF ZERO, SKIP THIS ONE
021736  026016  000000G         15$:    CMP     Z.NAM(R0),(SP)    ; DOES THE NAME MATCH?
021742  001403                          BEQ     40$               ; IF EQ, YES (C-BIT CLEAR)
021744  005302                  20$:    DEC     R2
021746  001371                          BNE     10$               ; LOOP $PDVNM TIMES
021750  000261                  30$:    SEC                       ; INDICATE FAILURE
021752  005226                  40$:    INC     (SP)+             ; PURGE STACK
021754  006016                          ROR     (SP)              ; SAVE C-BIT
021756  005741                          TST     -(R1)             ; CALCULATE PDV INDEX
021760  166701  000000G                 SUB     $PDVTA,R1         ;       ...
021764  006316                          ASL     (SP)              ; RESTORE C-BIT
021766                                  RETURN

                000001                  .END
```

```
VENA        CREATED BY  MAURO  ON 11-SEP-85 AT 12:08      PAGE 7
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE             REFERENCES

FR$EXC = 000000           #5-79
FR$FRM = 000010           #5-79
FR$FTL = 000005           #5-79
FR$OPN = 000004           #5-79
FR$RFD = 000006           #5-79
FR$SBU = 000012           #5-79
FR$SHO = 000003           #5-79
FR$UBU = 000013           #5-79
FR$UPT = 000014           #5-79
F.ADD    000016           #5-69       61-3451
F.CEV    000030           #5-69       6-161      45-2610    59-3310    62-3490    62-3492    62-3529    62-3573    62-3594
                           62-3623
F.CIR    000024           #5-69
F.CLS    000002           #5-69       45-2594    59-3343    60-3380    60-3398    61-3434    61-3439    62-3538    62-3578
                           62-3587    62-3636    63-3678    63-3684    72-3930
F.EVT    000004           #5-69       45-2609    62-3602    62-3605    62-3611
F.FLG    000014           #5-69       45-2601    59-3342    60-3387    60-3391    61-3443    63-3688    63-3696    63-3699
F.LEN    000016           #5-69       45-2593    45-2597    59-3307    60-3379    60-3382    61-3433    61-3436    62-3487
                           62-3616    63-3677    63-3680
F.LIN    000020           #5-69       45-2606    61-3455
F.LNK    000000           #5-69
F.MOD    000022           #5-69
F.NRBD = ****** GX        *27-1406   *27-1408   *28-1582   *28-1584   *39-2020   *40-2161   *42-2344   *42-2346   *42-2419
                          *42-2426   *42-2428   *45-2583   *62-3520   *62-3544   *62-3575   *62-3620   *62-3643   *82-4347
F.REM    000026           #5-69       59-3344    60-3393    61-3447    63-3701
F.RSIZ = ****** GX        *11-442    *20-919    *20-927    *21-977    *21-984    *22-1031   *22-1037   *26-1267   *26-1290
                          *26-1307   *26-1324   *26-1341   *26-1358   *27-1399   *27-1407   *28-1570   *28-1573   *28-1583
                          *29-1620   *30-1649   *32-1713   *36-1830   *36-1832   *37-1870   *37-1872   *38-1912   *39-1990
                          *39-1993   *39-2015   *39-2040   *39-2044   *39-2046   *40-2133   *40-2136   *40-2157   *41-2273
                          *42-2339   *42-2345   *42-2390   *42-2423   *42-2427   *45-2593   *49-2774   *50-2831   *53-3020
                          *59-3341   *60-3379   *61-3433   *62-3503   *62-3515   *62-3616   *63-3677   *74-4009   *74-4025
                          *74-4028   *74-4065   *75-4094   *75-4103   *75-4123   *76-4139   *76-4142   *77-4164   *78-4179
                          *81-4271   *81-4275   *82-4321   *82-4328   *84-4415
F.SEV    000004           #5-69       57-3177    62-3510    62-3526    62-3556    62-3591    62-3627
F.URBD = ****** GX        *27-1398   *27-1400   *28-1572   *28-1574   *42-2338   *42-2340   *42-2389   *42-2391   *45-2584
                          *53-3019   *53-3022   *63-3672   *63-3708
GCLASS   000116 R         #6-141
GETLEN   013572 R          48-2731    49-2767    50-2800   #51-2938
GETNXT   017102 R          62-3523    62-3621   #63-3672
GMASK    000106 R         #6-140      57-3203    57-3205    62-3497    62-3525    62-3564
GTSLT    006044 R          33-1738    34-1766   #35-1795
INIEVT   017454 R          59-3325    59-3336   #70-3870
LCLFLG   000123 R         #6-144     *74-4035    74-4045   *74-4068
LD.ALI = 000030           #5-73
LD.CEX = 000024           #5-73
LD.CIR = 000035           #5-73
LD.LIN = 000025           #5-73
LD.LOG = 000031           #5-73
LD.MOD = 000037           #5-73
LD.NOD = 000027           #5-73
LD.OBJ = 000032           #5-73
LD.PRO = 000026           #5-73
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 20
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE               REFERENCES

$VSEG      004020 RG     25-1234  #26-1290
$VSER      000242 RG     #11-393
$VTAD      001230 RG     #13-590
$XLINK   = ******  GX    27-1414   32-1720   41-2220   42-2354   72-3938
$$       = 177777          9-217     9-217    #9-217    #9-218     9-219     9-219    #9-219    #9-220     9-221
                           9-221    #9-221    #9-222     9-223     9-223    #9-223    #9-224     9-225     9-225
                          #9-225    #9-226     9-227     9-227    #9-227    #9-228     9-229     9-229    #9-229
                          #9-230     9-231     9-231    #9-231    #9-232     9-233     9-233    #9-233    #9-234
                           9-235     9-235    #9-235    #9-236     9-237     9-237    #9-237    #9-238     9-239
                           9-239    #9-239    #9-240     9-241     9-241    #9-241    #9-242     9-243     9-243
                          #9-243    #9-244     9-245     9-245    #9-245    #9-246     9-247     9-247    #9-247
                          #9-248     9-249     9-249    #9-249    #9-250     9-251     9-251    #9-251    #9-252
                           9-253     9-253    #9-253    #9-254     9-255     9-255    #9-255    #9-256     9-257
                           9-257    #9-257    #9-258     9-259     9-259    #9-259    #9-260     9-261     9-261
                          #9-261    #9-262     9-263     9-263    #9-263    #9-264     9-265     9-265    #9-265
                          #9-266     9-267     9-267    #9-267    #9-268     9-269     9-269    #9-269    #9-270
                           9-271     9-271    #9-271    #9-272     9-273     9-273    #9-273    #9-274     9-275
                           9-275    #9-275    #9-276     9-277     9-277    #9-277    #9-278     9-279     9-279
                          #9-279    #9-280     9-281     9-281    #9-281    #9-282     9-283     9-283    #9-283
                          #9-284     9-285     9-285    #9-285    #9-286     9-287     9-287    #9-287    #9-288
                           9-289     9-289    #9-289    #9-290     9-291     9-291    #9-291    #9-292     9-293
                           9-293    #9-293    #9-294     9-295     9-295    #9-295    #9-296     9-297     9-297
                          #9-297    #9-298     9-299     9-299    #9-299    #9-300     9-301     9-301    #9-301
                          #9-302     9-303     9-303    #9-303    #9-304     9-305     9-305    #9-305    #9-306
                           9-307     9-307    #9-307    #9-308     9-309     9-309    #9-309    #9-310     9-311
                           9-311    #9-311    #9-312     9-313     9-313    #9-313    #9-314   25-1218   25-1218
                        #25-1218  #25-1219
$$$      = 003632 R       #9-217     9-218    #9-219     9-220    #9-221     9-222    #9-223     9-224    #9-225
                           9-226    #9-227     9-228    #9-229     9-230    #9-231     9-232    #9-233     9-234
                          #9-235     9-236    #9-237     9-238    #9-239     9-240    #9-241     9-242    #9-243
                           9-244    #9-245     9-246    #9-247     9-248    #9-249     9-250    #9-251     9-252
                          #9-253     9-254    #9-255     9-256    #9-257     9-258    #9-259     9-260    #9-261
                           9-262    #9-263     9-264    #9-265     9-266    #9-267     9-268    #9-269     9-270
                          #9-271     9-272    #9-273     9-274    #9-275     9-276    #9-277     9-278    #9-279
                           9-280    #9-281     9-282    #9-283     9-284    #9-285     9-286    #9-287     9-288
                          #9-289     9-290    #9-291     9-292    #9-293     9-294    #9-295     9-296    #9-297
                           9-298    #9-299     9-300    #9-301     9-302    #9-303     9-304    #9-305     9-306
                          #9-307     9-308    #9-309     9-310    #9-311     9-312    #9-313     9-314  #25-1218
                         25-1219
$$$$     = 003656 R       #9-218    #9-220    #9-222    #9-224    #9-226    #9-228    #9-230    #9-232    #9-234
                          #9-236    #9-238    #9-240    #9-242    #9-244    #9-246    #9-248    #9-250    #9-252
                          #9-254    #9-256    #9-258    #9-260    #9-262    #9-264    #9-266    #9-268    #9-270
                          #9-272    #9-274    #9-276    #9-278    #9-280    #9-282    #9-284    #9-286    #9-288
                          #9-290    #9-292    #9-294    #9-296    #9-298    #9-300    #9-302    #9-304    #9-306
                          #9-308    #9-310    #9-312    #9-314  #25-1219
.ABTIM   = ******  GX    82-4321
.CLKHD   = ******  GX    82-4327
.MGMGE   = ******  GX    20-914    21-972    22-1026
.TLGTH   = ******  GX    39-1990   40-2133   76-4139   84-4415
.TSKHD   = ******  GX    84-4415
..COO    = ******  GX    39-2013   40-2155
```

```
	.TITLE	VEX2 - VNP EXPANSION ROUTINES
	.IDENT	/V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;	VNP - NORMAL CASE LINE TABLE BUILDER
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	27-FEB-78
;	VERSION 2.0 RELEASE
;
; 2.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;

```

```
                                          ;
                                          ; LIBRARY POINTER
                                          ;
001434                                    LIBR:
001434  112546                                    MOVB    (R5)+,-(SP)     ; ASSEMBLE DESCRIPTOR BIAS
001436  112566  000001                            MOVB    (R5)+,1(SP)     ;       ...
001442  012600                                    MOV     (SP)+,R0        ;       ...
001444  112546                                    MOVB    (R5)+,-(SP)     ; AND OFFSET
001446  112566  000001                            MOVB    (R5)+,1(SP)     ;       ...
001452  012602                                    MOV     (SP)+,R2        ;       ...
001454                                            CALL    $GETLB          ; READ IN LIBRARY BLOCK
001460  060201                                    ADD     R2,R1           ; POINT AT OUR DESCRIPTOR
001462  016124  000000G                           MOV     LB.ADR(R1),(R4)+ ; STORE LIBRARY ADDRESS
001466  005767  000000G                           TST     .MGMGE          ; MAPPED TARGET SYSTEM?
001472  100402                                    BMI     50$             ; IF MI, NO
001474  112524                                    MOVB    (R5)+,(R4)+     ; COPY USER MAPPING DISPLACEMENT
001476  112524                                    MOVB    (R5)+,(R4)+     ;       ...
001500  016701  000000G                   50$:    MOV     $LDA,R1         ; GET LIBRARY DATA ADDRESS WITHIN UNLOAD BLOCK
001504  010021                                    MOV     R0,(R1)+        ; STORE DESCRIPTOR BIAS
001506  010221                                    MOV     R2,(R1)+        ; AND OFFSET
001510  010167  000000G                           MOV     R1,$LDA         ; UPDATE LIBRARY DATA POINTER
001514                                            RETURN

                                          ;
                                          ; IF WE HAVE A PARTITION LINE TABLE, DO SOME SET UP HERE.
                                          ;
001516  005067  000004'                   PRLT:   CLR     FLAGA           ; FLAG FOR VECTOR BLOCK CODE ONLY
001522  005767  000000G                           TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM?
001526  100417                                    BMI     50$             ; NO IF MINUS
001530  026727  000000G 120000                    CMP     $SADDR,#120000  ; PARTITION LINE TABLE ALLOCATED?
001536  103413                                    BLO     50$             ; NO IF LOW!
001540  152767  000001  000004'                   BISB    #F.PRLT,FLAGA   ; SAY YES
001546  012767  000006' 000212'                   MOV     #ICBLK,ICBL     ; VECTOR BLOCK ADDR WITHIN VNP
001554  005067  000002'                           CLR     ICBPT           ; ZERO POINTER INTO ICB
001560  012767  177776  000000'                   MOV     #-2,NXTLN       ; INIT DISPLACEMENT POINTER
001566                                    50$:    RETURN

                                          ;
                                          ; IF A PARTITION LINE TABLE WAS ALLOCATED, SET UP ICB
                                          ; FOR EACH .INT HERE. WRITE THE BLOCK BEFORE EXITING VEX2.
                                          ;
001570  132767  000001  000004'           PRINT:  BITB    #F.PRLT,FLAGA   ; PARTITION LINE TABLE?
001576  001445                                    BEQ     100$            ; IF EQ, NO
001600  152767  000002  000004'                   BISB    #F.WRIT,FLAGA   ; WRITE ICB AT SOME POINT
001606  016700  000212'                           MOV     ICBL,R0         ; ADDR OF NEXT ICB ENTRY

                                                  .IF DF  R$$MPL                                          ;[MP01]
                                                  TST     ICBPT           ; FIRST INT ?                   ;[MP01]
                                                  BNE     5$              ; IF NO - BRANCH
                                                  CLR     (R0)+           ; CLEAR LINK WORK               ;[MP01]

                                          5$:
                                                  BIT     #DF.DAS,$DFLAG  ; D-SPACE SUPPORTED ?           ;[MP01]
                                                  BEQ     10$             ; IF NO - BRANCH                ;[MP01]
                                                  MOV     #13746,(R0)+    ; MOV @#KINAR5,-(SP)            ;[MP01]
                                                  MOV     #172372,(R0)+   ;  ...                          ;[MP01]

```

```
VEX2       CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 5
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME      REFERENCES

CALL              6-178     6-187     6-188     6-190     7-306     7-373    10-506    11-528    11-529    11-554
                 12-583    12-670    13-730    13-748
CALLR            13-761
CRASH             7-300
DHBDF$           #5-54      5-59
GETRV            #5-54      7-306     7-373
PUTAD            #5-54
PUTRC            #5-54     12-670    13-748
RAND             #7-306     7-306    #7-373     7-373   #12-670    12-670   #13-748    13-748
RESRG            #5-54      7-272    13-754
RETURN            6-191     7-203     7-213     7-215     7-223     7-232     7-238     7-246     7-252     7-276
                  7-279     7-295     7-310     7-319     7-325     7-340     7-342     7-353     7-366     7-376
                  7-383     7-395     7-397     7-410     7-412     7-420     7-422     7-430     7-432     7-439
                  8-471     9-487    10-507    10-514    11-537    11-571    12-594    12-608    12-660    12-708
                 13-767    13-774    13-780    13-783    14-791    14-796    14-801    14-806    14-811    14-816
                 14-821    14-826    14-831    14-836    14-841    15-848    15-853    15-858    15-863    15-868
SAVRG            #5-54      7-268    13-741
SLTDF$           #5-54      5-60
SWTCHI           #5-54      7-305     7-309
SWTCHO           #5-54     12-667    13-747    13-755
TTAB$            #5-65      5-93      5-94      5-95      5-96      5-97      5-98      5-99      5-100     5-101
                  5-102     5-103     5-104     5-105     5-106     5-107     5-108     5-109     5-110     5-111
                  5-112     5-113     5-114     5-115     5-116     5-117     5-118     5-119     5-120     5-121
                  5-122     5-123     5-124     5-125     5-126     5-127     5-128     5-129     5-130     5-131
                  5-132     5-133     5-134     5-135     5-136     5-137     5-138     5-139     5-140     5-141
                  5-142     5-143     5-144     5-145
```

```
VEX3      CREATED BY  MACRO  ON 29-JUN-85 AT 02:27      PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

CALL             6-156    8-202    8-221    10-253
CALLR            9-237
RETURN           6-158    7-169    7-175    7-181    7-186    8-203    8-222    10-254
TTAB$           #5-64     5-77     5-78     5-79     5-80     5-81     5-82     5-83     5-84     5-85
                 5-86     5-87     5-88     5-89     5-90     5-91     5-92     5-93     5-94     5-95
                 5-96     5-97     5-98     5-99     5-100    5-101    5-102    5-103    5-104    5-105
                 5-106    5-107    5-108    5-109    5-110    5-111    5-112    5-113    5-114    5-115
                 5-116    5-117    5-118    5-119    5-120    5-121    5-122    5-123    5-124    5-125
                 5-126    5-127    5-128    5-129
```

```
                                        ;
                                        ; SECONDARY LINKED FILE
                                        ;
001242  005267  000154'                 MXPTB:  INC     FLAGS           ; INDICATE MUX EXPANSION
001246  010377  000000G                         MOV     R3,@$ADRZ       ; VIRTUAL ADDRESS
001252  005077  000000G                         CLR     @$ADRX          ; CLEAR THE BIAS
001256  005767  000000G                         TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM?
001262  100413                                  BMI     5$              ; NO IF MINUS
001264  020327  120000                          CMP     R3,#120000      ; PARTITION LINE TABLE?
001270  103410                                  BLO     5$              ; NO IF LO
001272  010346                                  MOV     R3,-(SP)        ; LINE TALE ADDRESS
001274  042716  160000                          BIC     #160000,(SP)    ; CLEAR THE APR BITS
001300  012677  000000G                         MOV     (SP)+,@$ADRZ    ; NEW VIRTUAL ADDRESS
001304  016777  000000G 000000G                 MOV     $BIAS,@$ADRX    ; AND THE PROCSS BIAS
001312                                  5$:     GETAD   ,#10            ; ACCESS THE LINE TABLE
001324  012703  000000G                         MOV     #$BFR,R3        ; POINT TO CONTROLLER LINE TABLE
001330  005723                                  TST     (R3)+           ; POINT PAST FIRST LINE TABLE ADDRESS
001332  010346                                  MOV     R3,-(SP)        ; SAVE OUTPUT FILE POINTER
001334  112546                                  MOVB    (R5)+,-(SP)     ; GET REPEAT COUNT
001336  122525                                  CMPB    (R5)+,(R5)+     ; SKIP ALLOCATION ADDRESS
001340  016303  177776                          MOV     -2(R3),R3       ; GET FIRST LINE TABLE ADDRESS
001344  010546                                  MOV     R5,-(SP)        ; SAVE CURRENT BINARY FILE POINTER
001346  011605                          10$:    MOV     (SP),R5         ; RESTORE BINARY FILE POINTER
001350  126467  000013  000152'                 CMPB    L.UNT(R4),INDEX ; DO UNIT NUMBERS MATCH ?
001356  001003                                  BNE     20$             ; IF NE, NO
001360  026403  000004                          CMP     L.DDS(R4),R3    ; IS THE LINE TABLE POINTER CORRECT ?
001364  001026                                  BNE     101$            ; IF NE, NO .. ERROR
001366  005002                          20$:    CLR     R2              ; GET FUNCTION CODE
001370  152502                                  BISB    (R5)+,R2        ; ...
001372  122702  000000G                         CMPB    #.EOF,R2        ; IS IT END OF FILE ?
001376  001404                                  BEQ     30$             ; IF EQ, YES
001400  106302                                  ASLB    R2              ; ELSE, CONVERT TO A WORD INDEX
001402                                          CALL    @TABLE-2(R2)    ; EXPAND A FIELD
001406  000767                                  BR      20$             ; GO TILL END OF FILE
001410  005267  000152'                 30$:    INC     INDEX           ; TALLY ONE PASS THRU SECONDARY FILE
001414  126667  000002  000152'                 CMPB    2(SP),INDEX     ; ARE WE DONE ?
001422  001351                                  BNE     10$             ; IF NE, NO
001424  022626                                  CMP     (SP)+,(SP)+     ; ELSE, PURGE STACK
001426  012603                                  MOV     (SP)+,R3        ; RESTORE OUTPUT FILE POINTER
001430  005067  000152'                         CLR     INDEX           ; CLEAR SECONDARY INDEX FILE COUNT
001434  005067  000154'                         CLR     FLAGS           ; AND FLAGS
001440                                          RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
001442  000004                          101$:   IOT                     ; LINE TABLE POINTER (R3) IS SCREWED UP
                                        
                                        ;
                                        ; CURRENT TIME
                                        ;
001444                                  TIME:                           ; USE MIDNIGHT
                                        ;       CLR     R0              ; GET SECONDS/2 SINCE MIDNIGHT
                                        ;       CALL    ELTIME          ; ...
                                        ;       MOV     R3,@$ADRZ       ; VIRTUAL ADDRESS
                                        ;       CLR     @$ADRX          ; ASSUME NO BIAS
                                        ;       TST     .MGMGE          ; MAPPED SYSTEM ?
                                        ;       BMI     10$             ; IF MI, NO
```

```
        .TITLE  VFGDTE - INITIAL CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/
        .ENABL  LC                                                              ;[TD001]
                                                                                ;**-1
;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985  BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  04-MAR-82
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  8-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                        .SBTTL  BIASX - SET APR6 BIAS CORRECTLY

                                ;+
                                ;
                                ; BIASX - SET APR6 BIAS CORRECTLY
                                ;
                                ; INPUTS:
                                ;       R0 = ADDRESS (POOL OR APR5)
                                ;
                                ; OUTPUTS:
                                ;       R0 = ADDRESS (POOL OR APR6)
                                ;
                                ;-

000576                          BIASX:
000576  020027  120000                  CMP     R0,#120000              ; ADDRESS IN APR5 ?
000602  103404                          BLO     10$                     ; BR IF NO (ASSUME POOL)
000604  042700  160000                  BIC     #160000,R0              ; CLEAR ALL APR BITS
000610  052700  140000                  BIS     #140000,R0              ; AND BIAS FOR APR6
000614                          10$:    RETURN                          ; RETURN
```

```
                                        .TITLE  VFGPVC - INITIAL CONFIG FILE SCAN ACTION ROUTINES
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ; VNP - CONFIG FILE ACTION ROUTINES
                                ;
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  04-MAR-82
                                ;       DECNET-11M/S V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;
```

```
                                                .SBTTL  INIXCB - INITIALIZE X25 CIRCUIT BLOCK
                                        ;+
                                        ;
                                        ;
                                        ; INIXCB - INITIALIZE X25 CIRCUIT BLOCK
                                        ;
                                        ; INPUTS:
                                        ;       R0 - UNMAPPED ADDRESS OF CIRCUIT BLOCK
                                        ;       $PFLAG - PSI CHARACTERISTICS WORD                                       ;[TD001]
                                        ;
                                        ; OUTPUTS:
                                        ;       R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK
                                        ;       X25 CIRCUIT BLOCK INITIALIZED ACCORDING TO VALUES
                                        ;               SPECIFIED IN PVC$DF
                                        ;       R0,R1,R2 DESTROYED
                                        ;
                                        ;-
001062                                  INIXCB: CEACC$  R0,R3           ; GET MAPPED ADDRESS OF CIRCUIT BLOCK
001072  012701  000106                          MOV     #X$LEN,R1       ; GET SIZE OF CIRCUIT BLOCK
001076  032767  000000G 000000G                 BIT     #PF.PSV,$PFLAG  ; CHECK FOR PSV MODULE                  ;[TD001]
001104  001402                                  BEQ     5$              ; BRANCH IF NOT PRESENT                 ;[TD001]
001106  012701  000106                          MOV     #X$GLEN,R1      ; PSV NEEDS LONGER XCB                  ;[TD001]
001112                                  5$:     GETAD   R3,R1           ; READ IN CIRCUIT BLOCK                 ;[TD001]
001126  012702  000000G                         MOV     #$BFR,R2        ; GET CIRCUIT BLOCK ADDRESS             ;**-1
001132  105022                          10$:    CLRB    (R2)+           ; INITIALIZE CIRCUIT BLOCK TO ZEROES
001134  005301                                  DEC     R1              ; MORE TO INITIALIZE?
001136  003375                                  BGT     10$             ; BR IF YES
001140  010300                                  MOV     R3,R0           ; COPY VIRTUAL ADDRESS OF XCB
001142                                          CALL    BIASX           ; BIAS CORRECTLY FOR APR6
001146  016767  000016' 000020G                 MOV     PVCWSZ,X$WSZ+$BFR ; STORE WINDOW SIZE
001154  016767  000014' 000016G                 MOV     PVCPSZ,X$PKSZ+$BFR; STORE PACKET SIZE
001162  016767  000012' 000060G                 MOV     PVCCTM,X$CTIM+$BFR; STORE COUNTER TIMER
001170  016767  000000G 000024G                 MOV     DTEDES,X$DTE+$BFR ; STORE DTE DESCRIPTOR ADDRESS
001176  116767  000034' 000021G                 MOVB    PVCPRT,X$PRT+$BFR ; STORE PORT NUMBER
001204  016767  000010' 000026G                 MOV     PVCLCN,X$LCN+$BFR ; STORE LOGICAL CHANNEL NUMBER
001212  010067  000070G                         MOV     R0,X$TXQ+2+$BFR   ; INITIALISE TRANSMISSION QUEUE
001216  062767  000066  000070G                 ADD     #X$TXQ,X$TXQ+2+$BFR ; ...
001224  010067  000074G                         MOV     R0,X$RXQ+2+$BFR   ; INITIALISE RECEIVED DATA QUEUE
001230  062767  000072  000074G                 ADD     #X$RXQ,X$RXQ+2+$BFR ; ...
001236  010067  000064G                         MOV     R0,X$WAQ+2+$BFR   ; INITIALISE ACKNOWLEDGEMENT QUEUE
001242  062767  000062  000064G                 ADD     #X$WAQ,X$WAQ+2+$BFR ; ...
001250  010067  000100G                         MOV     R0,X$ABQ+2+$BFR   ; INITIALISE ABORTED DATA QUEUE
001254  062767  000076  000100G                 ADD     #X$ABQ,X$ABQ+2+$BFR ; ...
001262  116767  000062' 000005G                 MOVB    SP$P4,X$ST+$BFR   ; SET PORT IN DATA TRANSFER STATE
001270  112767  000001  000015G                 MOVB    #XT$PVC,X$TYP+$BFR ; SET CIRCUIT TYPE AS PVC
001276                                          PUTAD                     ; WRITE UPDATED CIRCUIT BLOCK
001302                                          RETURN
```

```
                                                                   I 15
VFGPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:30      PAGE 5
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE             REFERENCES

X$TYP      000015          *16-563
X$WAQ      000062          *16-558     16-559    *16-559
X$WSZ      000020          *16-548
Z.DAT    = ******   GX      20-731
Z.MAP    = ******   GX      20-728
Z.NAM    = ******   GX      20-723
$ALPHA   = 000022          #10-212
$ALPOL   = ******   GX      14-442
$ANY     = 000020          #10-212
$BFR     = ******   GX      14-450    *14-468    *14-469     15-495     15-498     15-502     16-542    *16-548    *16-549
                           *16-550    *16-551    *16-552    *16-553    *16-554    *16-555    *16-556    *16-557    *16-558
                           *16-559    *16-560    *16-561    *16-562    *16-563     17-588     17-599    *17-607    *17-608
                            20-732     20-733     20-734     20-737     20-739    *20-744    *20-745    *20-746    *20-747
                           *20-748
$BLANK   = 000006          #10-212
$CAT5    = ******   GX      12-378     19-647
$CEACC   = ******   GX      14-451     15-498     16-536     17-586
$DIGIT   = 000024          #10-212
$DNUMB   = 000014          #10-212
$EOS     = 000012          #10-212
$ERRN0     000064 R        #8-165      11-316     19-660
$ERRN1     000124 R        #8-166      12-384
$ERRN2     000160 R        #8-167      12-398
$ERRN3     000212 R        #8-168      13-421
$ERR00     000312 R        #8-170      12-369
$ERR05     000346 R        #8-171      11-344
$ERR09     000402 R        #8-172      11-332
$ERRP8     000256 R        #8-169      12-357
$ERR1T   = ******   GX       9-206
$EXIT    = 000000          #10-212
$FAIL    = 177777          #10-212
$GETAD   = ******   GX      14-452     14-463     15-501     16-541     17-587     17-598     20-736
$GETRV   = ******   GX      14-446     15-494     15-497     20-731
$GPRM    = ******           10-212
$LAMDA   = 000000          #10-212
$LLCTA   = ******   GX      20-710
$NUMBR   = 000002          #10-212
$PDVNM   = ******   GX      20-721
$PDVTA   = ******   GX      20-720
$PFLAG   = ******   GX      14-439     14-460     16-538
$PSIPT   = ******   GX      14-444     15-492
$PUTAD   = ******   GX      14-454     14-470     16-564     17-609     20-749
$QPVC      000000 RG       #9-190
$RADDR   = ******   GX     *14-446    *14-452    *14-463    *15-494    *15-497    *15-501    *16-541    *17-587    *17-598
                           *20-731    *20-736
$RAD50   = 000016          #10-212
$RBLCK   = ******   GX      13-410    *13-416     14-447    *14-455     15-491    *15-515     17-585    *17-610     20-730
                           *20-732    *20-750
$RONLY   = ******           10-212     10-212     10-212
$SAVAL   = ******   GX      13-409
$SLTA    = ******   GX      19-670     20-704
$SLTMA   = ******   GX      20-703     20-707
```

```

                                        ;
                                        ; PACKET SIZE
                                        ;
000236  005767  000000G                 SVPKT:  TST     .PNUMH                  ; LEGAL PACKET SIZE?
000242  001004                                  BNE     101$                    ; BR IF NO
000244  016767  000000G 000006'                 MOV     .PNUMB,PKTSVC           ; SAVE PACKET SIZE
000252                                          RETURN
                                        ;
                                        ; ERROR
                                        ;
000254                                  101$:   EMSG$R  3D                      ; ILLEGAL PACKET SIZE

                                        ;
                                        ; WINDOW SIZE
                                        ;
000262  005767  000000G                 SVWND:  TST     .PNUMH                  ; LEGAL WINDOW SIZE?
000266  001010                                  BNE     101$                    ; BR IF NO
000270  022767  000007  000000G                 CMP     #MAXWND,.PNUMB          ; ...
000276  103404                                  BLO     101$                    ; BR IF ILLEGAL WINDOW SIZE
000300  116767  000000G 000022'                 MOVB    .PNUMB,WNDSVC           ; SAVE WINDOW SIZE
000306                                          RETURN
                                        ;
                                        ; ERROR
                                        ;
000310                                  101$:   EMSG$R  3E                      ; ILLEGAL WINDOW SIZE

                                        ;
                                        ; FLAGS WORD
                                        ;
000316  005767  000000G                 SVFLAG: TST     .PNUMH                  ; LEGAL FLAGS WORD?
000322  001004                                  BNE     101$                    ; BR IF NO
000324  016767  000000G 000010'                 MOV     .PNUMB,FLGSVC           ; SAVE FLAGS WORD
000332                                          RETURN
                                        ;
                                        ; ERROR
                                        ;
000334                                  101$:   EMSG$R  3F                      ; ILLEGAL FLAGS WORD

                                        ;
                                        ; MAX RECALLS
                                        ;
000342  005767  000000G                 SVRCLL: TST     .PNUMH                  ; LEGAL MAX RECALLS ?
000346  001007                                  BNE     101$                    ; BR IF NO
000350  105767  000001G                         TSTB    .PNUMB+1                ; ...
000354  001004                                  BNE     101$                    ; BR IF ILLEGAL MAX RECALLS
000356  116767  000000G 000023'                 MOVB    .PNUMB,RCLSVC           ; SAVE RECALL TIMER
000364                                          RETURN
                                        ;
                                        ; ERROR
                                        ;
000366                                  101$:   EMSG$R  3G                      ; ILLEGAL MAX RECALLS
```

```
                                                .SBTTL  GETNXT - GET NEXT FILTER BLOCK

                                        ;+
                                        ;
                                        ; GETNXT - GET NEXT FILTER BLOCK IN LIST AND CHECK TO DETERMINE
                                        ;          WHETHER IT IS A QUALIFIED EVENT BLOCK WHICH IS
                                        ;          ASSOCIATED WITH THE GLOBAL EVENT BLOCK WHICH WAS
                                        ;          OPERATED ON
                                        ;
                                        ; INPUTS:
                                        ;       FLTBUF - ADDRESS OF NEXT BLOCK IN LIST
                                        ;
                                        ; OUTPUTS:
                                        ;       CARRY SET:
                                        ;          FLTBUF - ADDRESS OF NEXT EVENT FILTER BLOCK IN LIST
                                        ;                   THIS FILTER BLOCK CORRESPONDS TO THE SAME CLASS,
                                        ;                   SINK NODE, AND QUALIFIER AS THE GLOBAL FILTER
                                        ;                   BLOCK WHICH WAS OPERATED ON
                                        ;
                                        ;       CARRY CLEAR:
                                        ;          THE NEXT FILTER BLOCK IN THE LIST IS NOT ASSOCIATED
                                        ;          WITH THE GLOBAL FILTER BLOCK WHICH WAS OPERATED ON.
                                        ;
                                        ;-

017102                                  GETNXT: SWTCHI  #FLTBUF                 ; USE FLTBUF AS INPUT BUFFER
017110  016704  000170'                 10$:    MOV     FLTBUF,R4               ; GET POINTER TO NEXT BLOCK
017114  001475                                  BEQ     50$                     ; BR IF END OF LIST
017116                                          SAVRG   <R4>                    ; SAVE UNMAPPED ADDRESS
017120                                          CEACC$  R4                      ; CONVERT TO MAPPED ADDRESS
017130                                          GETAD   R4,#F.LEN               ; READ IN FILTER BLOCK
017146  016767  000002G 000000G                 MOV     F.CLS+$BFR,$RSIZE       ; GET LENGTH OF FILTER BLOCK
017154  042767  177700  000000G                 BIC     #^C<FF.MSK>,$RSIZE      ; ...
017162  022767  000016  000000G                 CMP     #F.LEN,$RSIZE           ; DID WE READ IN ENTIRE BLOCK?
017170  001402                                  BEQ     20$                     ; BR IF YES
017172                                          GETAD                           ; ELSE READ IN ENTIRE BLOCK
017176                                  20$:    RESRG   <R4>                    ; RESTORE UNMAPPED ADDRESS
017200  016746  000172'                         MOV     F.CLS+FLTBUF,-(SP)      ; GET EVENT CLASS AND LENGTH
017204  042716  000077                          BIC     #FF.MSK,(SP)            ; ISOLATE EVENT CLASS
017210  026726  000074'                         CMP     CLASS,(SP)+             ; SAME EVENT CLASS?
017214  001335                                  BNE     10$                     ; BR IF NO - CHECK NEXT BLOCK
017216  016746  000204'                         MOV     F.FLG+FLTBUF,-(SP)      ; GET FLAGS WORD
017222  042716  177770                          BIC     #^C<FF.MON!FF.CON!FF.FIL>,(SP) ; ISOLATE SINK TYPE
017226  016746  000120'                         MOV     EVQUAL,-(SP)            ; GET FLAGS WORD FOR GLOBAL BLOCK
017232  042716  177770                          BIC     #^C<FF.MON!FF.CON!FF.FIL>,(SP) ; ISOLATE SINK TYPE
017236  022626                                  CMP     (SP)+,(SP)+             ; MATCH ON SINK TYPE?
017240  001323                                  BNE     10$                     ; BR IF NO
017242  032767  100000  000120'                 BIT     #FF.REM,EVQUAL          ; QUALIFIED BY SINK NODE?
017250  001005                                  BNE     30$                     ; BR IF YES
017252  032767  100000  000204'                 BIT     #FF.REM,F.FLG+FLTBUF    ; THIS BLOCK QUALIFIED BY SINK NODE?
017260  001313                                  BNE     10$                     ; BR IF YES, CHECK NEXT BLOCK
017262  000410                                  BR      40$                     ; MATCH ON SINK NODES
017264  032767  100000  000204' 30$:    BIT     #FF.REM,F.FLG+FLTBUF    ; QUALIFIED BY SINK NODE?
017272  001706                                  BEQ     10$                     ; BR IF NO
017274  026767  000216' 000130'                 CMP     F.REM+FLTBUF,SINKND     ; DO SINK NODES MATCH?
017302  001302                                  BNE     10$                     ; BR IF NO
017304  000261                          40$:    SEC                             ; INDICATE BLOCK FOUND
```

```
   4044 020216  012703  000000G                      MOV     #$BFR,R3                        ; POINT AT BUFFER
   4045 020222  105767  000123'                      TSTB    LCLFLG                          ; Test first time flag
   4046 020226  001013                               BNE     30$                             ; If NE, skip vector init
   4047
   4048                                                                                      ; Initialize the vector
   4049 020230  010500                               MOV     R5,R0                           ; Get the number of PLBs
   4050 020232  010004                               MOV     R0,R4                           ; Calc virtual address of PLB
   4051 020234  006304                               ASL     R4                              ; ''
   4052 020236  160401                               SUB     R4,R1                           ; ''
   4053 020240  066604  000002                       ADD     2(SP),R4                        ; ''
   4054 020244  010423                       27$:    MOV     R4,(R3)+                        ; Init vector entry to point at PLB
   4055 020246  062704  000052                       ADD     #P$LEN,R4                       ; calc next PLB virtual address
   4056 020252  005300                               DEC     R0                              ; decrement PLB count
   4057 020254  003373                               BGT     27$                             ; loop till vector is initialized
   4058
   4059 020256  105023                       30$:    CLRB    (R3)+                           ; Zero the PLB's
   4060 020260  005301                               DEC     R1                              ;  until byte count runout
   4061 020262  003375                               BGT     30$                             ; ...
   4062
   4063 020264  012601                       35$:    MOV     (SP)+,R1                        ; Restore byte count
   4064 020266  012600                               MOV     (SP)+,R0                        ; and starting virtual address
   4065 020270                                       PUTRC   R0,R1                           ; WRITE BACK THE PLB DATA BASE
   4066 020304  022702  001000                       CMP     #512.,R2                        ; MORE TO TRANSFER?
   4067 020310  103005                               BHIS    40$                             ; BR IF NO
   4068 020312  105267  000123'                      INCB    LCLFLG                          ; make first time flag non-zero
   4069 020316  060100                               ADD     R1,R0                           ; Update starting virtual address
   4070 020320  010201                               MOV     R2,R1                           ; GET NUMBER OF BYTES REMAINING
   4071 020322  000725                               BR      10$                             ; BR TO TRANSFER NEXT GROUP
   4072
   4073 020324  000241                       40$:    CLC                                     ; INDICATE SUCCESS
   4074 020326                               50$:    RETURN
   4075
```

```
Symbol table

ACPNAM   000000R   002 CL$MAN=  000000        DL$HLT=  000000        D$RNN    000002        ERR49    003426R   002
AC$DNT=  000002        CL$PAZ=  034100        DL$IST=  000001        D$RTMR   000076        ERR5     000524R   002
AC$X25=  000001        CL$PLH=  034000        DL$MAI=  000004        D$SEG    000036        ERR6     000560R   002
AE$CIR=  000003        CL$PLL=  000600        DL$OFF=  000001        D$SER    000032        ERR7     000642R   002
AE$LIN=  000001        CL$PRT=  034200        DL$ON =  000000        D$SQRL   000052        ERR77    000664R   002
AE$MOD=  000004        CL$ROU=  010000        DL$RUN=  000003        D$$BUG=  177514        ERR8     000714R   002
ALPLB    020022R       CL$SES=  000200        DL$SHU=  000002        D$$ISK=  000000        ERR9     000764R   002
AT$ACL=  000100        CL$SGE=  035000        DL$SYN=  000005        D$$L11=  000001        EVQUAL   000120R   002
AT$ADN=  000040        CL$SSE=  035100        DSBLIN   003230R       D$$YNC=  000000        EVTVAL   000000R
AT$AUP=  000020        CL$TRN=  000400        DV.CCL=  000002        D$$YNM=  000000        EVTVSZ=  000020
AT$CYC=  000004        CL$XL2=  013700        DV.COM=  020000        D.NAM =  ****** GX     EV$ACF=  000201
AT$LV1=  000002        CL$XL3=  013600        DV.DIR=  000010        D.PCB =  ****** GX     EV$ADR=  000420
AT$LV2=  000001        CL$X2S=  013500        DV.EXT=  000400        D.UCB =  ****** GX     EV$ADU=  000417
AT$NEX=  000200        CM$SET=  ****** GX     DV.F11=  040000        EF$ACT=  000001        EV$APL=  000400
AT$UP =  000010        CNUDTB   003714R       DV.ISP=  002000        ENALIN   003134R       EV$ARC=  000421
A$CIR    000003        CONNAM   000160R   002 DV.MBC=  000400        ERR1     000336R   002 EV$AUC=  000010
A$LEN    000004        CURUNM   000164R   002 DV.MNT=  100000        ERR10    001040R   002 EV$AUS=  000003
A$NID    000000        CUSTEV=  000740        DV.MSD=  000100        ERR12    001104R   002 EV$CDF=  000520
A$TM     000003        C$$CKP=  000000        DV.OSP=  004000        ERR13    001136R   002 EV$CDZ=  000011
A$TMI    000002        C$$ORE=  000400        DV.PSE=  010000        ERR14    001176R   002 EV$DBR=  000302
A$TSZ    000000        C$$RSH=  177564        DV.REC=  000001        ERR15    001244R   002 EV$GAS=  035101
A$TYP    000002        C.AR5    000014        DV.SDI=  000020        ERR16    001314R   002 EV$HCE=  035114
A$$CHK=  000000        C.AST    000012        DV.SQD=  000040        ERR17    001342R   002 EV$HCI=  035113
A$$CPS=  000000        C.CSTP=  000012        DV.SWL=  001000        ERR18    001414R   002 EV$HFE=  000506
A$$PRI=  000000        C.DST    000016        DV.TTY=  000004        ERR19    001442R   002 EV$IFL=  000413
A$$TRP=  000000        C.EFN    000003        DV.UMD=  000200        ERR2     000372R   002 EV$IFO=  000415
A.ACC    000020        C.LGTH=  000020        D$AMXC   000072        ERR20    001472R   002 EV$IFS=  000414
A.LNK    000000        C.LNK    000000        D$AMXH   000074        ERR21    001520R   002 EV$INF=  000515
A.NAM    000002        C.MRKT=  000000        D$ANN    000000        ERR22    001572R   002 EV$LDL=  000407
A.PUD    000010        C.RQT    000002        D$BRPR   000102        ERR23    001636R   002 EV$LDN=  010416
A.REM    000012        C.RSI    000012        D$BRTM   000100        ERR24    001664R   002 EV$LDO=  000411
A.UCB    000010        C.SCHD=  000002        D$DELF   000045        ERR25    001726R   002 EV$LDS=  000410
BF.TRC=  000004        C.SRC    000014        D$DELW   000046        ERR26    001762R   002 EV$LSC=  000500
CHAN     000070R   002 C.SSHT=  000004        D$END =  000104        ERR27    002020R   002 EV$LUP=  000412
CHFIL    014240R       C.SUB    000012        D$FNB    000034        ERR28    002054R   002 EV$NOL=  000402
CHKEVT   017352R       C.SYST=  000006        D$HIOR   000024        ERR29    002110R   002 EV$NRC=  000416
CHKEXE   003730R       C.SYTK=  000010        D$HOST   000022        ERR3     000424R   002 EV$NSC=  000200
CHKXPT   021516R       C.TCB    000004        D$INAC   000044        ERR30    002144R   002 EV$NUL=  000401
CHNAM    014666R       C.TIM    000006        D$INCT   000042        ERR31    002176R   002 EV$NVR=  000406
CHSTA    014142R       C.UIC    000016        D$IPL    000051        ERR32    002236R   002 EV$OPL=  000403
CHTBLE   000262R   002 DEAFIL   017600R       D$LID    000020        ERR33    002306R   002 EV$PCC=  034000
CKCON    012772R       DEFCON   000044R   002 D$LNAM   000006        ERR34    002354R   002 EV$PCI=  034001
CKEVE    012144R       DEFDEV   000032R   002 D$LNUM   000014        ERR35    002402R   002 EV$PCM=  034002
CKFIL    013072R       DEFFIL   000020R   002 D$LST    000047        ERR36    002444R   002 EV$PFE=  000404
CKMDEV   013634R       DEFMON   000040R   002 D$MAXC   000064        ERR37    002476R   002 EV$PPC=  034003
CKMON    012676R       DEFTYP   000026R   002 D$MAXH   000066        ERR38    002546R   002 EV$RCF=  000517
CKNAME   012536R       DEFUIC   000036R   002 D$MAXV   000070        ERR39    002614R   002 EV$RDC=  010001
CKNODE   014070R       DEFUNT   000034R   002 D$MLL    000040        ERR4     000462R   002 EV$RDR=  010002
CKSINK   013746R       DEFVER   000030R   002 D$MNOD   000041        ERR40    002670R   002 EV$RJE=  035106
CLASS    000074R   002 DF.CRC=  000010        D$NA     000062        ERR41    002742R   002 EV$RSC=  000501
CLKQB    000050R   002 DF.FOC=  000020        D$NBEA   000056        ERR42    003000R   002 EV$RUL=  000405
CLRBIT   005724R       DF.LEN=  000030        D$NBRA   000054        ERR43    003050R   002 EV$SNA=  035000
CLREVT   017334R       DF.RAN=  000040        D$NEND=  000054        ERR44    003122R   002 EV$SNF=  000516
CL$ASZ=  010500        DF.UNK=  000100        D$NLN    000030        ERR45    003152R   002 EV$SPE=  035001
CL$DLL=  000500        DISMNT   020330R       D$NN     000060        ERR46    003226R   002 EV$XCE=  034110
CL$ECL=  000300        DLXPX    000125R   002 D$OUTT   000043        ERR47    003300R   002 EV$XDI=  013600
CL$LDN=  010400        DL$AST=  000002        D$RETF   000050        ERR48    003350R   002 EV$XGW=  034111
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 8
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

LF.ACT   = ******  GX         16-706     23-1067    24-1105    24-1115
LF.BRO   = ******  GX         11-454     17-776     17-784
LF.ENA   = ******  GX         11-484     11-496     23-1075    24-1131    24-1139
LF.LPB   = ******  GX         16-713     33-1742    34-1770
LF.MTP   = ******  GX         13-592     14-634     15-666
LF.RDY   = ******  GX         11-451     16-704     17-773     20-905     21-963     22-1017    81-4252
LF.SER   = ******  GX         12-542     12-550     12-556
LINFLT     012216 R           44-2500   #45-2582
LN.OFF   = ******  GX         24-1106    24-1112
LN.ON    = ******  GX         23-1068    23-1072
LO.ADD     000022            #5-73
LO.ADR     000052            #5-73
LO.CLS     000002            #5-73       46-2641   *56-3140    56-3146    59-3321
LO.CON     000006            #5-73
LO.COP     000010            #5-73       41-2233    41-2266
LO.COS     000020            #5-73       17-783     17-797
LO.CSR     000010            #5-73       5-73
LO.CTL     000032            #5-73
LO.DES     000002            #5-73       5-73
LO.DTE     000047            #5-73
LO.EVT     000004            #5-73       45-2560    56-3141    57-3194    57-3213    62-3491    62-3498    62-3511    62-3552
                              62-3555    62-3560    62-3567    62-3572    62-3601    69-3850    69-3851    69-3852    69-3853
LO.GRO     000024            #5-73
LO.HOS     000040            #5-73       78-4183
LO.HTM     000032            #5-73       80-4227
LO.INA     000062            #5-73       26-1342
LO.INC     000060            #5-73       26-1308
LO.LDT     000046            #5-73
LO.LID     000030            #5-73
LO.LIN     000004            #5-73
LO.LTM     000034            #5-73       79-4207
LO.MAC     000022            #5-73       14-640
LO.MDE     000024            #5-73       15-672
LO.NAM     000044            #5-73       27-1432    27-1441    27-1455    28-1514    28-1526    28-1535    28-1559
LO.NET   = 000002            #5-73
LO.NNM     000022            #5-73       54-3060
LO.OFL     000012            #5-73       41-2235    41-2241   *41-2246    41-2248    41-2265
LO.ONM     000004            #5-73       41-2251    41-2252    41-2267    41-2268
LO.OUT     000056            #5-73       26-1325
LO.OWN     000026            #5-73       16-696
LO.PAR     000010            #5-73
LO.PRI     000012            #5-73
LO.RET     000064            #5-73       26-1359
LO.RPA     000030            #5-73       36-1841
LO.SCR     000016            #5-73
LO.SEG     000054            #5-73       26-1291
LO.SKA     000014            #5-73
LO.SKN     000014            #5-73       53-3024
LO.SNM     000035            #5-73       47-2685    50-2821    50-2829    51-2938
LO.TPA     000020            #5-73       37-1879
LO.TRB     000034            #5-73       45-2603    71-3900
LO.TRI     000030            #5-73       13-597     13-602     13-607
```

```
VENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 21
MACRO CROSS REFERENCE                                    CREF   04.00
MACRO NAME          REFERENCES

ADJDF$              #5-62     5-83
ANBDF$              #5-60     5-76
ASL$                #5-62    #5-93      56-3139
ASR$                #5-103    46-2642
CALL                  11-397    11-401    11-407    11-412    11-440    11-442    11-489    11-502    11-505    11-509
                      11-514    11-515    11-516    11-517    11-520    11-521    11-522    13-594    13-599    13-604
                      13-608    13-614    13-615    14-637    14-641    14-646    14-647    15-669    15-673    15-678
                      15-679    16-697    16-701    16-752    16-753    16-754    16-755    16-756    17-802    17-803
                      20-919    20-927    20-940    20-941    21-977    21-984    21-995    21-996    22-1031   22-1037
                      22-1048   22-1049   24-1154   25-1176   25-1202   25-1217   26-1267   26-1290   26-1292   26-1307
                      26-1309   26-1324   26-1326   26-1341   26-1343   26-1358   26-1360   27-1386   27-1394   27-1399
                      27-1407   27-1414   27-1416   27-1421   27-1430   27-1434   27-1436   27-1442   27-1445   27-1465
                      27-1466   27-1467   28-1493   28-1499   28-1507   28-1518   28-1527   28-1531   28-1536   28-1540
                      28-1554   28-1569   28-1570   28-1573   28-1578   28-1580   28-1583   28-1591   28-1592   28-1593
                      28-1594   29-1619   29-1620   30-1648   30-1649   32-1712   32-1713   32-1715   32-1720   33-1738
                      34-1766   35-1799   36-1828   36-1830   36-1832   36-1853   37-1868   37-1870   37-1872   37-1891
                      38-1909   38-1912   38-1919   38-1925   38-1926   39-1943   39-1953   39-1966   39-1987   39-1990
                      39-1993   39-1998   39-1999   39-2003   39-2015   39-2017   39-2022   39-2040   39-2044   39-2046
                      39-2052   39-2056   39-2057   39-2058   39-2060   39-2062   39-2063   39-2064   39-2065   39-2066
                      39-2068   39-2069   39-2070   40-2086   40-2092   40-2098   40-2113   40-2131   40-2133   40-2136
                      40-2140   40-2141   40-2145   40-2157   40-2159   40-2166   40-2170   40-2171   40-2172   40-2174
                      40-2176   40-2177   40-2178   40-2179   40-2180   41-2208   41-2220   41-2222   41-2229   41-2253
                      41-2259   41-2272   41-2273   41-2276   41-2278   41-2283   41-2284   41-2285   42-2313   42-2315
                      42-2339   42-2345   42-2354   42-2356   42-2363   42-2368   42-2373   42-2374   42-2376   42-2384
                      42-2390   42-2397   42-2401   42-2405   42-2409   42-2413   42-2422   42-2423   42-2427   42-2436
                      42-2437   42-2438   43-2469   43-2471   43-2473   44-2491   44-2500   44-2512   44-2525   44-2534
                      45-2568   45-2582   45-2592   45-2593   45-2599   45-2620   46-2664   47-2698   47-2703   47-2706
                      47-2712   47-2713   48-2731   48-2737   49-2767   49-2771   49-2774   50-2800   50-2817   50-2827
                      50-2831   50-2846   50-2850   50-2868   50-2894   50-2912   52-2987   52-2995   53-3020   53-3029
                      53-3042   54-3065   54-3073   56-3125   56-3147   57-3173   57-3183   57-3204   57-3214   57-3216
                      57-3220   57-3223   57-3228   59-3312   59-3325   59-3327   59-3328   59-3331   59-3335   59-3336
                      59-3340   59-3341   59-3345   59-3347   59-3353   60-3378   60-3379   60-3384   61-3432   61-3433
                      61-3438   62-3493   62-3499   62-3502   62-3503   62-3512   62-3514   62-3515   62-3523   62-3527
                      62-3530   62-3534   62-3537   62-3540   62-3554   62-3557   62-3561   62-3566   62-3569   62-3574
                      62-3577   62-3580   62-3586   62-3589   62-3592   62-3595   62-3603   62-3607   62-3609   62-3613
                      62-3615   62-3616   62-3621   62-3625   62-3628   62-3632   62-3635   62-3638   63-3676   63-3677
                      63-3682   65-3754   72-3938   72-3940   74-4003   74-4009   74-4013   74-4016   74-4020   74-4025
                      74-4028   74-4031   74-4065   75-4090   75-4094   75-4097   75-4101   75-4103   75-4107   75-4115
                      75-4116   75-4123   76-4139   76-4142   76-4144   77-4164   78-4179   78-4188   79-4205   79-4208
                      79-4210   80-4225   80-4228   80-4230   81-4271   81-4275   82-4306   82-4321   82-4328   82-4333
                      82-4344   82-4346   82-4348   82-4355   82-4356   83-4375   83-4391   83-4393   83-4396   84-4415
                      84-4422
CALLR               #5-62     39-1945   39-1949   39-1963   39-1975   39-1986
CEACC$              #5-61     28-1569   29-1619   30-1648   32-1712   41-2272   42-2422   45-2592   59-3340   60-3378
                      61-3432   62-3502   62-3514   62-3537   62-3577   62-3586   62-3615   62-3635   63-3676
CLKDF$              #5-59     5-67
CTRDF$              #5-62     5-68
DDCDF$              #5-61     5-81
DHBDF$              #5-62     5-84
DMPDF$              #5-62     5-80
ECDDB$              #5-59     5-70
ERBLK$              #5-59     9-218     9-220     9-222     9-224     9-226     9-228     9-230     9-232     9-234
```

```
                                  .MCALL  SWTCHI,SWTCHO,PUTRC,GETRV,DHBDF$,SLTDF$,RESRG,SAVRG,PUTAD

                         ;**
                         ; LOCAL MACROS
                         ;**
        000000                            DHBDF$          ; DEFINE DECNET HOME BLOCK OFFSETS
        000000                            SLTDF$          ; DEFINE SLT OFFSETS

                         ;
                         ; DEFINE TRANSFER TABLE ENTRY
                         ;
                                  .MACRO  TTAB$   AA,BB
                                  .WORD   AA
                                  .ENDM   TTAB$

                         ;**
                         ; LOCAL DATA
                         ;**

        000000                    .PSECT  DATA,D

                         ;
                         ; VARIABLES FOR VECTOR BLOCK ROUTINES
                         ;
        000000  000000   NXTLN:   .WORD   0               ; USED AS AN OFFSET FOR ICB DISPLACEMENT
        000002  000000   ICBPT:   .WORD   0               ; FUDGER
                000012   JSROF=12                         ; OFFSET TO JSR IN LINE TABLE

        000004  000000   FLAGA:   .WORD   0               ; FLAG FOR VECTOR BLOCK ROUTINES
                000001   F.PRLT=1                         ; PARTITION LINE TABLE ALLOCATED
                000002   F.WRIT=2                         ; WRITE VECTOR BLOCK TO IMAGE

        000006           ICBLK:   .BLKW   66.             ; ROOM FOR PLENTY OF ENTRIES
        000212  000006'  ICBL:    .WORD   ICBLK           ; POINTER TO ICBLK ENTRY
        000214           KRBB:    .BLKB   60              ; INTERNAL KRB COPY

                         ;
                         ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                         ;
        000274           TABLE:   TTAB$   ADDRB,NULL      ; EXPANSION ROUTINE IS FIRST ENTRY
        000276                    TTAB$   ADDRW,NULL      ; CLEANUP ROUTINE IS SECOND ENTRY
        000300                    TTAB$   BLKB,POP1
        000302                    TTAB$   BLKW,POP1
        000304                    TTAB$   BYTE,POP1
        000306                    TTAB$   CORE,XCORE
        000310                    TTAB$   CSR,POP1
        000312                    TTAB$   CTIM,NULL
        000314                    TTAB$   DVCHA,NULL
        000316                    TTAB$   EOF,NULL
        000320                    TTAB$   FILE,POP1
        000322                    TTAB$   INT,POP2
        000324                    TTAB$   LIBR,XLIBR
        000326                    TTAB$   LINKS,NULL
        000330                    TTAB$   LFILE,POP3
        000332                    TTAB$   LSTHD,NULL
        000334                    TTAB$   LTAB,NULL
        000336                    TTAB$   MPTAB,POP3
```

```
                                  10$:
                                           .ENDC                                                     ;[MP01]

001612  012720  013746                     MOV     #13746,(R0)+       ; MOV @#KISAR5,-(SP) CODE        ;[MP01]
001616  012720  172352                     MOV     #172352,(R0)+      ; ...                            ;[MP01]

001622  012720  012737                     MOV     #12737,(R0)+       ; MOV $BIAS,@#KISAR5 CODE
001626  016720  000000G                    MOV     $BIAS,(R0)+        ; ...
001632  012720  172352                     MOV     #172352,(R0)+      ; ...

                                           .IF DF  R$$MPL                                            ;[MP01]

                                           BIT     #DF.DAS,$DFLAG     ; D-SPACE SUPPORTED              ;[MP01]
                                           BEQ     20$                ; IF NO - BRANCH                 ;[MP01]
                                           MOV     #12737,(R0)+       ; MOV $BIAS,@#KINAR5             ;[MP01]
                                           MOV     $BIAS,(R0)+        ; ...                            ;[MP01]
                                           MOV     #172372,(R0)+      ; ...                            ;[MP01]
                                  20$:
                                           .ENDC                                                     ;[MP01]

001636  062767  000002  000000'            ADD     #2,NXTLN           ; UP THE POINTER
001644  016701  000000'                    MOV     NXTLN,R1           ; NEXT ENTRY IN $VECT1
001650  016702  000000G                    MOV     $SADDR,R2          ; LINE TABLE ADDRESS
001654  066102  000000G                    ADD     $VECT1(R1),R2      ; POINT AT CORRECT .INT CODE
001660  062702  000012                     ADD     #JSROF,R2          ; POINT AT JSR WITHIN .INT
001664  012720  000137                     MOV     #137,(R0)+         ; JMP @#LINE-TABLE-ADDRESS
001670  010220                             MOV     R2,(R0)+           ; ...
001672  010067  000212'                    MOV     R0,ICBL            ; READY FOR NEXT ENTRY
001676  016761  000002' 000000G            MOV     ICBPT,$VECT1(R1)   ; SET UP FOR $TMVCT
001704  066767  000000G 000002'            ADD     $ICBLN,ICBPT       ; FOR NEXT TIME
001712                            100$:    RETURN


                                  ;
                                  ; SEE IF AN ICB SHOULD BE WRITTEN TO THE IMAGE.
                                  ;
001714  132767  000002  000004'   WRVBL:   BITB    #F.WRIT,FLAGA      ; IF SET THEN DO A WRITE
001722  001413                             BEQ     100$               ; NO IF EQ
001724                                     SWTCHO  #ICBLK             ; SET UP BUFFER ADDRESS
                                           .IF NDF R$$MPL                                            ;[MP01]

001732                                     PUTRC   $ICBAD,$ICBSZ      ; WRITE ICB TO IMAGE
                                           .IFF                                                      ;[MP01]
                                           TST     .MPRO              ; MULTIPROCESSOR ?               ;[MP01]
                                           BPL     10$                ; IF YES - BRANCH                ;[MP01]

                                           SUB     #2,$ICBSZ          ; UPDATE ICB SIZE                ;[MP01]
                                           PUTRC   $ICBAD,$ICBSZ      ; WRITE ICB TO IMAGE             ;[MP01]
                                           ADD     #2,$ICBSZ          ; RESTORE ICB SIZE               ;[MP01]
                                           BR      100$               ; CONTINUE                       ;[MP01]

                                  10$:     MOV     $SLTA,R5           ; GET SLT ADDRESS                ;[MP01]
                                           MOV     L.KRBA(R5),R5      ; GET KRB ADDRESS                ;[MP01]
                                           SWTCHI  #KRBB              ; SET INPUT BUFFER               ;[MP01]
                                           ADD     #K.PRM,R5          ; ADJUST FOR NEGATIVE OFFSETS    ;[MP01]

                                           GETRV   R5,#$KRBLN         ; READ KRB                       ;[MP01]
                                           MOV     KRBB,R5            ; GET ICB ADDRESS/2 WITH CPU NUMBER ;[MP01]
```

```
                                                                            J  8
**FILE**ID**VEX3







VV      VV   EEEEEEEEEE   XX      XX     333333
VV      VV   EEEEEEEEEE   XX      XX     333333
VV      VV   EE           XX      XX   33      33
VV      VV   EE           XX      XX   33      33
VV      VV   EE             XX  XX             33
VV      VV   EE             XX  XX             33
VV      VV   EEEEEEEE         XX             33
VV      VV   EEEEEEEE         XX             33
VV      VV   EE             XX  XX             33
VV      VV   EE             XX  XX             33
  VV  VV     EE           XX      XX   33      33       ....
  VV  VV     EE           XX      XX   33      33       ....
    VV       EEEEEEEEEE   XX      XX     333333         ....
    VV       EEEEEEEEEE   XX      XX     333333         ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                              J 9
**FILE**ID**VEX5








VV      VV   EEEEEEEEEE   XX      XX   5555555555
VV      VV   EEEEEEEEEE   XX      XX   5555555555
VV      VV   EE           XX      XX   55
VV      VV   EE           XX      XX   55
VV      VV   EE             XX  XX     555555
VV      VV   EE             XX  XX     555555
VV      VV   EEEEEEEE         XX             55
VV      VV   EEEEEEEE         XX             55
VV      VV   EE             XX  XX             55
VV      VV   EE             XX  XX             55
  VV  VV     EE           XX      XX   55      55         ....
  VV  VV     EE           XX      XX   55      55         ....
    VV       EEEEEEEEEE   XX      XX     555555           ....
    VV       EEEEEEEEEE   XX      XX     555555           ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                 ;       CMP     R3,#120000       ; PARTITION LINE TABLE ?
                                 ;       BLO     10$              ; IF LO, NO
                                 ;       MOV     R3,-(SP)         ; GET LINE TABLE ADDRESS
                                 ;       BIC     #160000,(SP)     ; WITHOUT APR SELECT BITS
                                 ;       MOV     (SP)+,@$ADRZ     ; SET NEW VIRTUAL ADDRESS
                                 ;       MOV     $BIAS,@$ADRX     ; SET BIAS OF PROCESS
                                 ;10$:   GETAD   ,#2              ; ACCESS LINE TABLE
                                 ;       MOV     R0,$BFR          ; STORE TIME
                                 ;       PUTAD                    ; RE-WRITE LINE TABLE
001444  005723                           TST     (R3)+            ; UPDATE POINTER
001446                                   RETURN

                                 ;
                                 ; DEAD POLLING RATIO BYTE
                                 ; ACTIVE POLLING RATIO (APR)
                                 ;
001450                           DPRB:
001450  005203                   APR:    INC     R3               ; SKIP IT
001452                                   RETURN

                                 ;
                                 ; DEAD POLLING RATIO WORD
                                 ;
001454  005723                   DPRW:   TST     (R3)+            ; SKIP IT
001456                                   RETURN
```

```
                        .SBTTL  MACRO DEFINITIONS

                ;
                ; LIBRARY MACROS
                ;
                        .MCALL  EMSG$R,SLTDF$,SAVRG,RESRG,DTEDF$,GETAD,CEACC$
                        .MCALL  PHBDF$,NTLER$,ISTAT$,TRAN$,STATE$,ASL$,GETRV
                ;
                ; LIBRARY SYMBOLS
                ;
 000000                 DTEDF$                          ; LOCAL DTE DESCRIPTOR BLOCK OFFSETS
 000000                 PHBDF$                          ; DEFINE PSI HOME BLOCK OFFSETS
 000000                 SLTDF$                          ; DEFINE SLT OFFSETS
                ;
                ; LOCAL MACRO DEFINITIONS
                ;
                        .MACRO  SAVMAP
                        MOV     $RBLCK,-(SP)
                        .ENDM

                        .MACRO  RESMAP
                        MOV     (SP)+,$RBLCK
                        .ENDM
```

```
                                        .SBTTL  LINE-ID ACTION ROUTINES
                                        .ENABL  LSB
                                ;
                                ; DEVICE NAME
                                ;
000616  022767  000003  000000G DEVNAM: CMP     #3,.PSTCN               ; VALID DEVICE NAME?
000624  103425                          BLO     101$                    ; BR IF NO
000626  016700  000000G                 MOV     .PSTPT,R0               ; GET ADDRESS OF FIRST CHARACTER
000632                                  CALL    $CAT5                   ; CONVERT TO RAD50
000636  010167  000042'                 MOV     R1,LINNAM               ; SAVE DEVICENAME
000642                                  RETURN
                                ;
                                ; DEVICE CONTROLLER NUMBER
                                ;
000644  005767  000000G         DEVCTL: TST     .PNUMH                  ; LEGAL CONTROLLER NUMBER VALUE?
000650  001013                          BNE     101$                    ; BR IF NO
000652  016767  000000G 000044'         MOV     .PNUMB,LINCTL           ; SAVE CONTROLLER NUMBER
000660                                  RETURN
                                ;
                                ; DEVICE UNIT NUMBER
                                ;
000662  005767  000000G         DEVUNT: TST     .PNUMH                  ; LEGAL UNIT NUMBER VALUE?
000666  001004                          BNE     101$                    ; BR IF NO
000670  116767  000000G 000045'         MOVB    .PNUMB,LINUNT           ; SAVE UNIT NUMBER
000676                                  RETURN
                                ;
                                ; ERRORS
                                ;
000700                          101$:   EMSG$R  06                      ; ILLEGAL LINE-ID
                                        .DSABL  LSB
```

```
     54                                                    .SBTTL  MACRO DEFINITIONS
     55
     56                                            ;
     57                                            ; LIBRARY MACROS
     58                                            ;
     59                                                    .MCALL  EMSG$R,SAVRG,RESRG,PVCDF$,XCBDF$,GETAD,PUTAD,GETRV
     60                                                    .MCALL  PHBDF$,NTLER$,STATE$,TRAN$,ISTAT$,CEACC$,PLBDF$
     61                                                    .MCALL  SLTDF$,XPDDB$,CTRDF$,DTEDF$
     62                                            ;
     63                                            ; LIBRARY SYMBOLS
     64                                            ;
     65 000000                                             DTEDF$                          ; LOCAL DTE DESCRIPTOR OFFSETS
     66 000000                                             CTRDF$                          ; TRANSPORT COUNTER BLOCK OFFSETS
     67 000000                                             XPDDB$                          ; XPT'S DDB OFFSETS
     68 000000                                             PHBDF$                          ; PSI HOME BLOCK OFFSETS
     69 000000                                             PVCDF$                          ; PVC NAME BLOCK OFFSETS
     70 000000                                             SLTDF$                          ; SLT OFFSETS
     71 000000                                             XCBDF$                          ; X25 CIRCUIT CONTROL BLOCKS
     72 000000                                             PLBDF$                          ; PLB OFFSETS
     73
     74                                            ;
     75                                            ;
     76                                            ; LOCAL MACRO DEFINITIONS
     77                                            ;
     78                                                    .MACRO  SAVMAP
     79                                                    MOV     $RBLCK,-(SP)
     80                                                    .ENDM
     81
     82                                                    .MACRO  RESMAP
     83                                                    MOV     (SP)+,$RBLCK
     84                                                    .ENDM
     85                                            ;
     86                                            ; SET UP APR5 MAPPING
     87                                            ;
     88                                                    .MACRO  MAP     BIAS
     89                                                    MOV     BIAS,$RBLCK
     90                                                    .ENDM
     91                                            ;
     92                                            ; SET UP APR5 BIAS
     93                                            ;
     94                                                    .MACRO  BIAS    REG
     95                                                    BIC     #160000,REG     ; CLEAR CURRENT MAPPING BITS
     96                                                    BIS     #120000,REG     ; SET UP APR5 MAPPING
     97                                                    .ENDM
```

```
                                                         .SBTTL  FNDHSH - FIND HASH TABLE ENTRY

                                                ;+
                                                ;
                                                ; FNDHSH - FIND HASH TABLE ENTRY
                                                ;
                                                ; INPUTS:
                                                ;        PVCLCN - LOGICAL CHANNEL NUMBER
                                                ;        PVCPRT - PORT NUMBER
                                                ;        HSHADD - APR6 ADDRESS OF HASH TABLE ENTRY
                                                ;
                                                ; OUTPUTS:
                                                ;        CARRY CLEAR - TABLE ENTRY AVAILABLE AND SET UP WITH LCN/PORT
                                                ;        CARRY SET - NO ENTRIES AVAILABLE
                                                ;        R0,R1,R2 DESTROYED
                                                ;        MAPPING PRESERVED
                                                ;
                                                ;-
001304                                          FNDHSH: SAVMAP                          ; SAVE MAPPING
001310                                                  CEACC$  DTEDES,R2               ; CONVERT TO MAPPED ADDRESS
001322                                                  GETAD   R2,#L$LEN               ; READ IN DTE DESCRIPTOR
001340  016767  000030G 000000G                         MOV     L$CHTB+$BFR,HSHADD ; GET HASH TABLE ADDRESS
001346  005002                                          CLR     R2                      ; INITIAL OFFSET FOR HASH TABLE INSERTION
001350  016700  000010'                                 MOV     PVCLCN,R0               ; GET LOGICAL CHANNEL NUMBER
001354  016701  000000G                         10$:    MOV     HSHSZ,R1                ; GET NUMBER OF ENTRIES IN HASH TABLE
001360  005401                                          NEG     R1                      ; GET MASK FOR HIGH BITS
001362  040100                                          BIC     R1,R0                   ; COMPUTE ENTRY NUMBER (MODULO TABLE SIZE)
001364  010001                                          MOV     R0,R1                   ; SAVE ENTRY NUMBER
001366  006300                                          ASL     R0                      ; MULTIPLY BY 4 TO DETERMINE OFFSET
001370  006300                                          ASL     R0                      ; ... INTO HASH TABLE
001372  066700  000000G                                 ADD     HSHADD,R0               ; DETERMINE ENTRY ADDRESS IN HASH TABLE
001376                                                  GETAD   R0,#4                   ; READ IN  HASH TABLE ENTRY
001414  005767  000002G                                 TST     $BFR+2                  ; IS THIS ENTRY AVAILABLE?
001420  001407                                          BEQ     20$                     ; BR IF YES (CLEARS CARRY)
001422  005202                                          INC     R2                      ; ELSE, COLLISION - UPDATE OFFSET
001424  026702  000000G                                 CMP     HSHSZ,R2                ; ANY ENTRIES AVAILABLE?
001430  103413                                          BLO     30$                     ; BR IF NO (SETS CARRY)
001432  060201                                          ADD     R2,R1                   ; CALCULATE ENTRY NUMBER + OFFSET
001434  010100                                          MOV     R1,R0                   ; SAVE ENTRY NUMBER + OFFSET
001436  000746                                          BR      10$                     ; FIND NEXT AVAILABLE ENTRY
001440  016767  000010' 000000G 20$:                    MOV     PVCLCN,$BFR             ; STORE LOGICAL CHANNEL NUMBER IN HASH TABLE
001446  116767  000034' 000002G                         MOVB    PVCPRT,$BFR+2           ; STORE PORT NUMBER IN LOW BYTE OF HASH TABLE
001454                                                  PUTAD                           ; WRITE UPDATED HASH TABLE ENTRY
001460                                          30$:    RESMAP                          ; RESTORE MAPPING
001464                                                  RETURN
```

```
VFGPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:30       PAGE 6

SYMBOL CROSS REFERENCE                                     CREF   04.00

SYMBOL  VALUE               REFERENCES

$STRNG  = 000004            #10-212
$SUBXP  = 000010            #10-212
$TRIB   = ******  GX         19-678
$X3DBS  = ******  GX         11-308
$X3DWS  = ******  GX         11-309
$$HTIM  = ******  GX         13-406
$$LTIM  = ******  GX         13-407
$$$FLG  = 177777            #10-212
$$$KEY  = 177777            #10-212
.PNUMB  = ******  GX         11-323    11-339    12-391    12-393    19-665    19-668    19-671    19-677    19-678
                             19-682
.PNUMH  = ******  GX         11-321    11-337    12-389    19-666    19-680
.PSTCN  = ******  GX         12-374
.PSTPT  = ******  GX         12-376
.TPARS  = ******  GX          9-197
```

```
                                 ;
                                 ; RETRY TIMER
                                 ;
000374  005767  000000G          SVRRTM: TST     .PNUMH                  ; LEGAL RECALL TIMER ?
000400  001004                           BNE     101$                    ; BR IF NO
000402  016767  000000G 000012'          MOV     .PNUMB,RTMSVC           ; SAVE RECALL TIMER
000410                                   RETURN
                                 ;
                                 ; ERROR
                                 ;
000412                           101$:   EMSG$R  3H                      ; ILLEGAL RECALL TIMER

                                 ;
                                 ; NO DTE ADDRESS SPECIFIED
                                 ;
000420  032767  000002  000010'  NODTE:  BIT     #SF$INC,FLGSVC          ; IS THIS AN INCOMING SVC?
000426  001401                           BEQ     101$                    ; BR IF NO
000430                                   RETURN
                                 ;
                                 ; ERROR
                                 ;
000432                           101$:   EMSG$R  3M                      ; DTE ADDRESS MISSING
                                 ;
                                 ; START OF DTE ADDRESS
                                 ;
000440  012700  000017           SVDTES::MOV     #DTELEN,R0              ; GET LENGTH OF REMOTE DTE ADDRESS
000444  012701  000025'                  MOV     #DTESVC,R1              ; GET ADDRESS OF START OF BUFFER
000450  105021                   10$:    CLRB    (R1)+                   ; INITIALIZE BUFFER
000452  005300                           DEC     R0                      ; MORE TO INITIALIZE?
000454  003375                           BGT     10$                     ; BR IF YES
000456  105067  000024'                  CLRB    COUNT                   ; INITIALIZE DIGIT COUNT
000462  012767  000025' 000000'          MOV     #DTESVC,NEXT            ; START AT BEGINNING OF BUFFER
000470                                   CALL    STRNXT                  ; STORE CHARACTER
000474                                   RETURN

                                 ;
                                 ; PROCESS DTE DIGIT
                                 ;
000476  026727  000000' 000044'  SVDTED: CMP     NEXT,#DTEEND            ; IS ADDRESS TOO LONG?
000504  101003                           BHI     101$                    ; BR IF YES
000506                                   CALL    STRNXT                  ; STORE CHARACTER
000512                                   RETURN
                                 ;
                                 ; ERRORS
                                 ;
000514                           101$:   EMSG$R  3I                      ; ILLEGAL DTE ADDRESS

                                 ;
                                 ; END OF DTE ADDRESS
                                 ;
000522  004567  000000G          SVDTEE:JSR      R5,$SAVRG               ; SAVE R3-R5
000526  012700  000010                   MOV     #<DTELEN+1>/2,R0        ; GET LENGTH OF PACKED ADDRESS
000532  012704  000025'                  MOV     #DTESVC,R4              ; POINT TO NUMBER TO CONVERT
000536  012705  000044'                  MOV     #DTEPCK,R5              ; POINT TO STORAGE FOR PACKED NUMBER
000542                                   CALL    PCKBCD                  ; PACK REMOTE DTE ADDRESS IN BCD
000546                                   RETURN
```

```
 3704 017306  000401                    BR      60$                     ; FINISH UP
 3705
 3706 017310  000241            50$:    CLC                             ; INDICATE END OF EVENT CLASS LIST
 3707
 3708 017312                    60$:    SWTCHI                          ; RESTORE $BFR AS INPUT BUFFER
 3709 017320                            RETURN
```

```
  4077                                          .SBTTL  DISMNT - DISMOUNT THE ACP
  4078                                  ;+
  4079                                  ; *** - DISMNT - DISMOUNT THE ACP
  4080                                  ;
  4081                                  ; INPUTS:
  4082                                  ;       NONE.
  4083                                  ; OUTPUTS:
  4084                                  ;       NONE.
  4085                                  ;
  4086                                  ;-
  4087
  4088 020330                           DISMNT:
  4089 020330  012700  000010'                  MOV     #NTINAM,R0              ; GET RAD50 TASK NAME ADDRESS
  4090 020334                                   CALL    TSKSCH                  ; READ TASK TCB
  4091
  4092 020340  016700  000012G                  MOV     T.RCVL+$BFR,R0          ; POINT AT THE SEND DATA PACKET
  4093 020344  001414                           BEQ     15$                     ; IF EQ, ACP NOT MOUNTED
  4094 020346                                   GETRV   R0,#36.                 ; READ IN PACKET
  4095
  4096 020364  042767  010000  000006G          BIC     #LS.CXO,LR.STS+6+$BFR   ; ELSE CLEAR SET EXE STATE ON
  4097 020372                                   PUTRC                           ; REWRITE PACKET
  4098                                  ;
  4099                                  ; DE-ALLOCATE THE PLB DATA BASE
  4100                                  ;
  4101 020376                           15$:    CALL    CHKXPT                  ; GET XPT'S DATA BASE ADDRESS
  4102 020402  103435                           BCS     20$                     ; IF CS, ERROR
  4103 020404                                   GETRV   R2,#N$XLEN              ; READ IN XPT'S DATA BASE
  4104 020422  016700  000016G                  MOV     N$PLD+$BFR,R0           ; GET NUMBER OF PLB'S ENTRIES
  4105 020426  001423                           BEQ     20$                     ; IF EQ, NONE
  4106 020430  012701  000052                   MOV     #P$LEN,R1               ; ELSE, GET LENGTH OF EACH PLB
  4107 020434                                   CALL    $MUL                    ; CALCULATE SIZE OF PLB DATA BASE
  4108 020440  016700  000016G                  MOV     N$PLD+$BFR,R0           ; Get number of channels again
  4109 020444  006300                           ASL     R0                      ; multiply by two for vector size
  4110 020446  060001                           ADD     R0,R1                   ; add to total allocation
  4111 020450  016700  000020G                  MOV     N$PLD+2+$BFR,R0         ; GET ADDRESS OF ALLOCATION
  4112 020454  001410                           BEQ     20$                     ; IF EQ, NONE
  4113 020456  005067  000016G                  CLR     N$PLD+$BFR              ; CLEAR THE DATA BASE POINTERS
  4114 020462  005067  000020G                  CLR     N$PLD+2+$BFR            ; ...
  4115 020466                                   PUTRC                           ; REWRITE THE XPT DATA BASE
  4116 020472                                   CALL    $DEA16                  ; DEALLOCATE THE PLB DATA BASE
  4117
  4118
  4119 020476                           20$:
  4120 020476  032767  000000G 000000G          BIT     #VF.SS,$VFLAG           ; 11S ?
  4121 020504  001413                           BEQ     30$                     ; IF NO - BRANCH
  4122 020506  012767  000340  000000G          MOV     #340,$BFR               ; REPLACE LOCATION 2
  4123 020514                                   PUTRC   #2,#2                   ; ...
  4124 020534                           30$:    RETURN
```

```
Symbol table

EV$XMX= 000514        FF.ADD= 000020            F.MOD   000022            LO.COS  000020        LS.ALI= 000016
EV$XRS= 000512        FF.CIR= 000040            F.NRBD= ******  GX        LO.CSR  000010        LS.CEX= 000001
EV$XSC= 000513        FF.CON= 000001            F.REM   000026            LO.CTL  000032        LS.CIR= 000033
EV$X2S= 013500        FF.FIL= 000002            F.RSIZ= ******  GX        LO.DES  000002        LS.CXO= 010000
EV.CCB= 000001        FF.HST= 040000            F.SEV   000004            LO.DTE  000047        LS.CX5= 000400
EV.CIR= 000020        FF.LIN= 000010            F.URBD= ******  GX        LO.EVT  000004        LS.ECH= 001000
EV.LCB= 000100        FF.MOD= 000100            GCLASS  000116R       002 LO.GRO  000024        LS.FDX= 004000
EV.LIN= 000004        FF.MON= 000004            GETLEN  013572R           LO.HOS  000040        LS.HDX= 010000
EV.MAP= 000002        FF.MSK= 000077            GETNXT  017102R           LO.HTM  000032        LS.HLP= 000012
EV.MOD= 000040        FF.PRT= 000200            GMASK   000106R       002 LO.INA  000062        LS.LIN= 000003
EV.NOD= 000010        FF.QUL= 000370            GTSLT   006044R           LO.INC  000060        LS.LMC= 000007
EV.PRT= 000200        FF.REM= 100000            G$$TPP= 000000            LO.LDT  000046        LS.LOG= 000020
E$DATA  000020        FILDEV  000146R       002 G$$TSS= 000000            LO.LID  000030        LS.MOD= 000036
E$EVTS  000000        FILLEN= 000020            G$$TTK= 000000            LO.LIN  000004        LS.NOD= 000014
E$LCN   000016        FILNAM  000134R       002 G$$WRD= 000000            LO.LTM  000034        LS.NTI= 000013
E$LIN   000000        FILOPR  015772R           INIEVT  017454R           LO.MAC  000022        LS.OBJ= 000022
E$MOD   000012        FILPRO  014422R           I$$RAR= 000000            LO.MDE  000024        LS.OPT= 000400
E$NBR   000014        FILTYP  000142R       002 I$$RDN= 000000            LO.NAM  000044        LS.PRO= 000002
E$NBS   000020        FILUIC  000152R       002 I.OFF = 000024            LO.NET= 000002        LS.TOP= 002000
E$NCR   000034        FILUNT  000150R       002 K$$CNT= 177546            LO.NNM  000022        LS.TST= 000010
E$NCS   000036        FILVER  000144R       002 K$$CSR= 177546            LO.OFL  000012        LS.UNF= 020000
E$NIC   000044        FINFLT  015340R           K$$LDC= 000000            LO.ONM  000004        LS.XIT= 000011
E$NLEN  000050        FINOBJ  005376R           K$$TPS= 000074            LO.OUT  000056        LX.ALI= 000017
E$NLLA  000012        FINREM  005500R           LCLFLG  000123R       002 LO.OWN  000026        LX.CEX= 000004
E$NLNK  000000        FLTBUF  000170R       002 LD$LP = 000000            LO.PAR  000010        LX.CIR= 000034
E$NML   000040        FNDCTR  021022R           LD.ALI= 000030            LO.PRI  000012        LX.LIN= 000006
E$NMR   000024        FNDDCB  020616R           LD.CEX= 000024            LO.RET  000064        LX.LOG= 000021
E$NMS   000030        FNDDMP  002732R           LD.CIR= 000035            LO.RPA  000030        LX.NOD= 000015
E$NNOD  000002        FNDFIL  015534R           LD.LIN= 000025            LO.SCR  000016        LX.OBJ= 000023
E$NOD   000010        FNDFR   002136R           LD.LOG= 000031            LO.SEG  000054        LX.PRO= 000005
E$NRT   000042        FNDLIN  002304R           LD.MOD= 000037            LO.SKA  000014        L$$ASG= 000000
E$NRTP  000005        FNDPCL  002526R           LD.NOD= 000027            LO.SKN  000014        L$$DRV= 000000
E$NSEG  000010        FNDTRI  002310R           LD.OBJ= 000032            LO.SNM  000035        L$$P11= 000001
E$NTIM  000046        FNPID   002222R           LD.PRO= 000026            LO.TPA  000020        L$$11R= 000000
E$NUSE  000004        FR$BCC= 000007            LE.NRH= 000200            LO.TRB  000034        L.BABL  000062
E$PORT  000014        FR$CCF= 000001            LE.NRO= 000001            LO.TRI  000030        L.BUFU  000052
E$PRM   000002        FR$CDF= 000002            LE.NSH= 000002            LO.UNT  000033        L.CHA   000034
E$STAT  000006        FR$DAO= 000011            LE.NSO= 000020            LO.VCT  000014        L.CNTL  000024
E$STRT  000006        FR$EXC= 000000            LE.RCO= 000040            LO.VER  000042        L.COST= ******  GX
E$TCB   000004        FR$FRM= 000010            LE.SAE= 000004            LR.ACC  000020        L.CRC   000030
E$$XPR= 000000        FR$FTL= 000005            LE.STT= 000010            LR.ACO  000052        L.CRS   000040
E.CTL   000020        FR$OPN= 000004            LE.XTU= 000100            LR.ADD  000012        L.DISL  000042
E.DATA  000046        FR$RFD= 000006            LF.ACT= ******  GX        LR.ALI  000002        L.DISR  000044
E.EVT   000002        FR$SBU= 000012            LF.BRO= ******  GX        LR.BLK  000010        L.DLC = ******  GX
E.LCN   000042        FR$SHO= 000003            LF.ENA= ******  GX        LR.CTL  000004        L.DLM = ******  GX
E.LEN   000216        FR$UBU= 000013            LF.LPB= ******  GX        LR.DES  000010        L.DLS = ******  GX
E.LIN   000024        FR$UPT= 000014            LF.MTP= ******  GX        LR.HLP  000002        L.ENA   000066
E.LNK   000000        F$$LVL= 000001            LF.RDY= ******  GX        LR.LIN  000002        L.FLG = ******  GX
E.MOD   000036        F.ADD   000016            LF.SER= ******  GX        LR.NAM  000012        L.LNUM  000046
E.NOD   000034        F.CEV   000030            LINFLT  012216R           LR.PAS  000041        L.MPF = ******  GX
E.PDV   000021        F.CIR   000024            LN.OFF= ******  GX        LR.PRO  000002        L.NAST  000065
E.PORT  000040        F.CLS   000002            LN.ON = ******  GX        LR.STS  000000        L.NCUB  000012
E.[illegible] 000026  F.EVT   000004            LO.ADD  000022            LR.TRI  000006        L.NCUL= 000050
E.PVC   000044        F.FLG   000014            LO.ADR  000052            LR.TYP  000002        L.NIP   000010
E.SIZ   000022        F.LEN   000016            LO.CLS  000002            LR.UCB  000016        L.NLSE  000054
E.TIME  000004        F.LIN   000020            LO.CON  000006            LR.UIC  000020        L.NMST= ******  GX
FATAL = ******  GX    F.LNK   000000            LO.COP  000010            LR.UNT  000005        L.NRSE  000053
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 9
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE          REFERENCES

LO.UNT     000033       #5-73
LO.VCT     000014       #5-73
LO.VER     000042       #5-73     38-1916
LR.ACC     000020       #5-73
LR.ACO     000052       #5-73     42-2382   42-2410
LR.ADD     000012       #5-73     26-1269   28-1498   28-1530   28-1539   28-1564
LR.ALI     000002       #5-73     42-2301   42-2395
LR.BLK     000010       #5-73
LR.CTL     000004       #5-73
LR.DES     000010       #5-73     42-2367   42-2370   42-2399
LR.HLP     000002       #5-73
LR.LIN     000002       #5-73     5-73      5-73      5-73      5-73      5-73      5-73
LR.NAM     000012       #5-73     26-1271   26-1273   26-1275   27-1380   27-1450   28-1492   28-1506
LR.PAS     000041       #5-73     42-2380   42-2406
LR.PRO     000002       #5-73
LR.STS     000000       #5-73     39-1994  *39-1997  *39-2008  *39-2009  *39-2012   40-2137  *40-2139  *40-2150
                       *40-2151  *40-2154  *75-4096  *76-4143
LR.TRI     000006       #5-73
LR.TYP     000002       #5-73     41-2197   41-2264
LR.UCB     000016       #5-73     42-2398
LR.UIC     000020       #5-73     42-2378   42-2402
LR.UNT     000005       #5-73
LS.ALI   = 000016       #5-73
LS.CEX   = 000001       #5-73     39-2008   40-2150
LS.CIR   = 000033       #5-73
LS.CXO   = 010000       #5-73     39-1994   39-1997   39-2008   75-4096
LS.CX5   = 000400       #5-73     40-2137   40-2139   40-2150   76-4143
LS.ECH   = 001000       #5-73
LS.FDX   = 004000       #5-73
LS.HDX   = 010000       #5-73
LS.HLP   = 000012       #5-73
LS.LIN   = 000003       #5-73
LS.LMC   = 000007       #5-73
LS.LOG   = 000020       #5-73
LS.MOD   = 000036       #5-73
LS.NOD   = 000014       #5-73
LS.NTI   = 000013       #5-73
LS.OBJ   = 000022       #5-73
LS.OPT   = 000400       #5-73
LS.PRO   = 000002       #5-73
LS.TOP   = 002000       #5-73
LS.TST   = 000010       #5-73
LS.UNF   = 020000       #5-73     39-2009   39-2012   40-2151   40-2154
LS.XIT   = 000011       #5-73
LX.ALI   = 000017       #5-73
LX.CEX   = 000004       #5-73
LX.CIR   = 000034       #5-73
LX.LIN   = 000006       #5-73
LX.LOG   = 000021       #5-73
LX.NOD   = 000015       #5-73
LX.OBJ   = 000023       #5-73
LX.PRO   = 000005       #5-73
```

```
VENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 22
MACRO CROSS REFERENCE                                    CREF   04.00
MACRO NAME      REFERENCES

                 9-236       9-238       9-240       9-242       9-244       9-246       9-248       9-250       9-252       9-254
                 9-256       9-258       9-260       9-262       9-264       9-266       9-268       9-270       9-272       9-274
                 9-276       9-278       9-280       9-282       9-284       9-286       9-288       9-290       9-292       9-294
                 9-296       9-298       9-300       9-302       9-304       9-306       9-308       9-310       9-312       9-314
                #11-396      25-1219
ERMSG$          #5-59        9-217       9-219       9-221       9-223       9-225       9-227       9-229       9-231       9-233
                 9-235       9-237       9-239       9-241       9-243       9-245       9-247       9-249       9-251       9-253
                 9-255       9-257       9-259       9-261       9-263       9-265       9-267       9-269       9-271       9-273
                 9-275       9-277       9-279       9-281       9-283       9-285       9-287       9-289       9-291       9-293
                 9-295       9-297       9-299       9-301       9-303       9-305       9-307       9-309       9-311       9-313
                #11-396      25-1218
ERRPT$          #5-59       #11-396      11-514      11-515      11-516      11-517      11-520      11-521      11-522      13-614
                 13-615      14-646      14-647      15-678      15-679      16-752      16-753      16-754      16-755      16-756
                 17-802      17-803      20-940      20-941      21-995      21-996      22-1048     22-1049     24-1154     25-1217
                 27-1465     27-1466     27-1467     28-1591     28-1592     28-1593     28-1594     38-1925     38-1926     39-2056
                 39-2057     39-2058     39-2062     39-2063     39-2064     39-2065     39-2069     39-2070     40-2092     40-2170
                 40-2171     40-2172     40-2176     40-2177     40-2178     40-2179     41-2283     41-2284     41-2285     42-2313
                 42-2363     42-2436     42-2437     42-2438     43-2469     44-2534     46-2664     47-2712     47-2713     53-3042
                 54-3073     57-3228     59-3353     79-4210     80-4230     82-4356     83-4393     83-4396
EVLDF$          #5-61        5-79
FLTDF$          #5-61        5-69
GETAD           #5-60        20-919      20-927      21-977      21-984      22-1031     22-1037     29-1620     30-1649     32-1713
                 39-2040     45-2593     45-2599     60-3379     60-3384     61-3433     61-3438     63-3677     63-3682     81-4275
GETRV           #5-59        11-442      26-1267     26-1290     26-1307     26-1324     26-1341     26-1358     27-1399     28-1573
                 36-1830     36-1832     37-1870     37-1872     38-1912     39-1990     39-1993     40-2133     40-2136     42-2339
                 42-2390     49-2774     50-2831     53-3020     74-4009     74-4025     74-4028     75-4094     75-4103     76-4139
                 76-4142     77-4164     78-4179     81-4271     82-4321     82-4328     82-4333     82-4344     84-4415     84-4422
NSFDF$          #5-60        5-77
OBJDF$          #5-60        5-75
PCLDF$          #5-61        5-82
PLBDF$          #5-61        5-72
PUTAD           #5-60        13-608      14-641      15-673      27-1434     28-1518     28-1540     28-1570     32-1715     39-2046
                 41-2253     41-2273     42-2423     45-2620     59-3341     62-3503     62-3515     62-3540     62-3580     62-3589
                 62-3616     62-3638     79-4208     80-4228
PUTRC           #5-59        11-509      26-1292     26-1309     26-1326     26-1343     26-1360     27-1407     28-1583     36-1853
                 37-1891     38-1919     39-1998     39-2003     39-2015     39-2044     40-2140     40-2145     40-2157     42-2345
                 42-2427     74-4013     74-4031     74-4065     75-4097     75-4115     75-4123     76-4144     78-4188     82-4346
                 82-4348
RAND            #11-442      11-442     #11-509      11-509     #13-608      13-608     #14-641      14-641     #15-673      15-673
                #20-919      20-919     #20-927      20-927     #21-977      21-977     #21-984      21-984     #22-1031     22-1031
                #22-1037     22-1037    #26-1267     26-1267    #26-1290     26-1290    #26-1292     26-1292    #26-1307     26-1307
                #26-1309     26-1309    #26-1324     26-1324    #26-1326     26-1326    #26-1341     26-1341    #26-1343     26-1343
                #26-1358     26-1358    #26-1360     26-1360    #27-1399     27-1399    #27-1407     27-1407    #27-1434     27-1434
                #28-1518     28-1518    #28-1540     28-1540    #28-1570     28-1570    #28-1573     28-1573    #28-1583     28-1583
                #29-1620     29-1620    #30-1649     30-1649    #32-1713     32-1713    #32-1715     32-1715    #36-1830     36-1830
                #36-1832     36-1832    #36-1853     36-1853    #37-1870     37-1870    #37-1872     37-1872    #37-1891     37-1891
                #38-1912     38-1912    #38-1919     38-1919    #39-1990     39-1990    #39-1993     39-1993    #39-1998     39-1998
                #39-2003     39-2003    #39-2015     39-2015    #39-2040     39-2040    #39-2044     39-2044    #39-2046     39-2046
                #40-2133     40-2133    #40-2136     40-2136    #40-2140     40-2140    #40-2145     40-2145    #40-2157     40-2157
                #41-2253     41-2253    #41-2273     41-2273    #42-2339     42-2339    #42-2345     42-2345    #42-2390     42-2390
                #42-2423     42-2423    #42-2427     42-2427    #45-2593     45-2593    #45-2599     45-2599    #45-2620     45-2620
                #49-2774     49-2774    #50-2831     50-2831    #53-3020     53-3020    #59-3341     59-3341    #60-3379     60-3379
```

```
    111 000340                                  TTAB$   MXTAB,POP1
    112 000342                                  TTAB$   NOD,NULL
    113 000344                                  TTAB$   PRI,NULL
    114 000346                                  TTAB$   PRTOV,POP1
    115 000350                                  TTAB$   SCOM,XSCOM
    116 000352                                  TTAB$   SECSR,NULL
    117 000354                                  TTAB$   SLNB,NULL
    118 000356                                  TTAB$   SLNW,NULL
    119 000360                                  TTAB$   STNB,NULL
    120 000362                                  TTAB$   STNW,NULL
    121 000364                                  TTAB$   UNB,NULL
    122 000366                                  TTAB$   UNW,NULL
    123 000370                                  TTAB$   WORD,POP2
    124 000372                                  TTAB$   APR,POP1
    125 000374                                  TTAB$   DPRB,POP1
    126 000376                                  TTAB$   DPRW,POP2
    127 000400                                  TTAB$   TIME,POP2
    128 000402                                  TTAB$   MXPTB,XMXPTB
    129 000404                                  TTAB$   PECHA,NULL
    130 000406                                  TTAB$   X3CH01,NULL
    131 000410                                  TTAB$   X3CH02,NULL
    132 000412                                  TTAB$   X3CH03,NULL
    133 000414                                  TTAB$   X3CH04,NULL
    134 000416                                  TTAB$   X3CH05,NULL
    135 000420                                  TTAB$   X3CH06,NULL
    136 000422                                  TTAB$   X3CH07,NULL
    137 000424                                  TTAB$   X3CH08,NULL
    138 000426                                  TTAB$   X3CH09,NULL
    139 000430                                  TTAB$   X3CH10,NULL
    140 000432                                  TTAB$   X3CH11,NULL
    141 000434                                  TTAB$   X2CH01,NULL
    142 000436                                  TTAB$   X2CH02,NULL
    143 000440                                  TTAB$   X2CH03,NULL
    144 000442                                  TTAB$   X2CH04,NULL
    145 000444                                  TTAB$   X2CH05,NULL
    146                                 ;
    147                                 ; SECONDARY FILE INDEX VALUE
    148                                 ;
    149 000446                          INDEX:  .BLKW   1
    150
    151                                 ;
    152                                 ; MUX/MTP FLAGS
    153                                 ;
    154 000450                          FLAGS:  .BLKW   1               ; 100000=MUX, 200=MTP
    155
    156 000000                                  .PSECT
```

```
                                MOV     R5,R2           ; COPY ICB ADDRESS/2 WITH CPU NUMBER    ;[MP01]
                                BIC     #7777,R2        ; GET CPU NUMBER                        ;[MP01]
                                BIC     R2,R5           ; CLEAR CPU BITS                        ;[MP01]

                                SWAB    R2              ; ...                                   ;[MP01]
                                ASR     R2              ; ...                                   ;[MP01]
                                ASR     R2              ; ...                                   ;[MP01]
                                ASR     R2              ; ...                                   ;[MP01]
                                ASL     R5              ; GET ACTUAL ICB ADDRESS                ;[MP01]

                                MOV     R5,$ICBAD       ; COPY ICB ADDRESS                      ;[MP01]
                                ADD     .K6TAB,R2       ; GET ADDRESS OF THIS CPU PARTITION BIAS;[MP01]
                                SWTCHI                  ; SET INPUT BUFFER                      ;[MP01]
                                GETRV   R2,#2           ; READ BIAS                             ;[MP01]
                                SWTCHO  #ICBLK          ; SET UP BUFFER ADDRESS                 ;[MP01]

                                MOV     $BFR,$RBLCK     ; STORE BIAS FOR ICB WRITE              ;[MP01]
                                PUTAD   $ICBAD,$ICBSZ   ; WRITE ICB TO IMAGE                    ;[MP01]
                                SWTCHO                  ; SET OUTPUT BUFFER TO DEFAULT          ;[MP01]
                                .ENDC

    001752              100$:   RETURN
```

VEX3 - VNP EXPANSION ROUTINES C MACRO V05.03b Saturday 29-Jun-85 02:27
Table of contents

```
     5-   56     MACRO DEFINITIONS
     6-  133     $VEX3 - TEMPLATE EXPANSION CLEANUP
     7-  160     CLEANUP ROUTINES
```

Table of contents

```
     5-   56    MACRO DEFINITIONS
     6-  157    $VEX5 - TEMPLATE EXPANSION
     7-  200    EXPANSION ROUTINES
    11-  676    X3CH
    12-  734    X2CH
```

```
                                                ;+
                                                ; FLT16 - FIND A 16 BIT ALLOCATION LINE TABLE
                                                ;
                                                ; INPUTS:
                                                ;         R3=LINE TABLE ADDRESS
                                                ;         R4=PDV INDEX OF PROCESS
                                                ;
                                                ; OUTPUTS:
                                                ;         R0,R1,R2,R3=DESTROYED
                                                ;         UBIAS,UBA=UNLOAD BLOCK ADDRESS
                                                ;         BLOCK=LOCAL COPY OF UNLOAD BLOCK
                                                ;-
001460                                          FLT16:
001460  005067  000160'                         FLT:    CLR     UBA             ; INDICATE NO UNLOAD BLOCK FOUND YET
001464  012702  000164'                                 MOV     #BLOCK,R2       ; USE R2 AS BASE REGISTER
001470  012767  000100  000000G                         MOV     #64.,$RSIZE     ; SIZE OF UNLOAD BLOCK IN BYTES
001476                                                  SWTCHI  R2              ; USE BLOCK AS INPUT BUFFER
001502  005077  000000G                                 CLR     @$ADRZ          ; SET UP ADDRESS
001506  016777  000000C 000000G                         MOV     .CXUNL+$NTLHB,@$ADRX ; OF UNLOAD BLOCK
001514  001463                                          BEQ     100$            ; NO UNLOAD BLOCKS

001516                                          10$:    GETAD                   ; READ IN IMAGE OF UNLOAD BLOCK
001522  005722                                          TST     (R2)+           ; SKIP POINTER TO NEXT UNLOAD BLOCK
001524  005722                                  20$:    TST     (R2)+           ; POINT PAST PDV INDEX
001526  126204  177776                                  CMPB    -2(R2),R4       ; IS THIS THE RIGHT PDV ?
001532  001013                                          BNE     30$             ; IF NE, NO
001534  022203                                          CMP     (R2)+,R3        ; IS THIS OUR LINE TABLE?
001536  001440                                          BEQ     80$             ; IF EQ, YES
001540  101011                                          BHI     40$             ; IF HI, NO
001542  011246                                          MOV     (R2),-(SP)      ; EXTRACT SIZE FIELD
001544  042716  000003                                  BIC     #3,(SP)         ; ...
001550  066216  177776                                  ADD     -2(R2),(SP)     ; ADD TABLE ENTRY
001554  022603                                          CMP     (SP)+,R3        ; IS THIS OUR LINE TABLE?
001556  101030                                          BHI     80$             ; IF HI, YES
001560  000401                                          BR      40$             ; ELSE, SKIP ANY SUB-ALLOCATIONS
001562  005722                                  30$:    TST     (R2)+           ; SKIP THE LINE TABLE ADDRESS
001564  032722  000001                          40$:    BIT     #1,(R2)+        ; ANY SUB-ALLOCATIONS THIS ENTRY?
001570  001410                                          BEQ     50$             ; IF EQ, NO
001572  005046                                          CLR     -(SP)           ; GET NUMBER OF ALLOCATIONS
001574  152216                                          BISB    (R2)+,(SP)      ; ...
001576  005046                                          CLR     -(SP)           ; AND NUMBER OF LIBRARIES
001600  152216                                          BISB    (R2)+,(SP)      ; ...
001602  062616                                          ADD     (SP)+,(SP)      ; ADD THEM TOGETHER
001604  006316                                          ASL     (SP)            ; MULTIPLY BY 4
001606  006316                                          ASL     (SP)            ; ...
001610  062602                                          ADD     (SP)+,R2        ; GIVING NEXT TABLE ENTRY
001612  022702  000264'                         50$:    CMP     #BLOCK+64.,R2   ; IS IT PHYSICAL END OF BLOCK?
001616  001402                                          BEQ     60$             ; IF EQ, YES
001620  005712                                          TST     (R2)            ; IS IT LOGICAL END OF BLOCK?
001622  001340                                          BNE     20$             ; IF NE, NO - CHECK NEXT ENTRY
001624  012702  000164'                         60$:    MOV     #BLOCK,R2       ; RESET R2 TO START OF BLOCK
                                                ;       CLR     @$ADRZ          ; SET NEW ADDRESS; THIS CAUSES BUG;
001630  011277  000000G                                 MOV     (R2),@$ADRX     ; SET NEW ADDRESS
001634  001330                                          BNE     10$             ; READ IN BLOCK
001636  000412                                          BR      100$            ; DONE

001640  005742                                  80$:    TST     -(R2)           ; RESET R2 TO FIRST WORD OF ENTRY
```

```
    79                                            .SBTTL  LOCAL SYMBOL DEFINITIONS
    80
    81                                    ;
    82                                    ; LOCAL SYMBOL DEFINITIONS
    83                                    ;
    84          000040                            SPACE = 40                ; ASCII SPACE
    85          000040                            HSHMN  = 32.              ; MINIMUM VALUE FOR HASH TABLE SIZE
    86          001000                            HSHMX  = 512.             ; MAXIMUM VALUE FOR HASH TABLE SIZE
    87          000020                            NUMDTE = 16.              ; MAXIMUM DTES ALLOWED
    88
    89                                    ;
    90                                    ; SAVED REGISTER OFFSETS ON STACK FOR $SAVAL
    91                                    ;
    92          000002                            R$R0 = 2                  ; SAVED R0
    93          000004                            R$R1 = 4                  ; SAVED R1
    94          000006                            R$R2 = 6                  ; SAVED R2
    95          000010                            R$R3 = 10                 ; SAVED R3
    96          000012                            R$R4 = 12                 ; SAVED R4
    97          000014                            R$R5 = 14                 ; SAVED R5
```

```
                                                .SBTTL  STRNXT - STORE NEXT DIGIT

                                        ;+
                                        ;
                                        ; STRNXT - STORE NEXT DIGIT
                                        ;
                                        ; INPUTS:
                                        ;       .PCHAR - MATCHED CHARACTER FROM TPARS TRANSITION
                                        ;       NEXT - ADDRESS OF NEXT BYTE AVAILABLE FOR STORAGE
                                        ;
                                        ; OUTPUTS:
                                        ;       NEXT - ADDRESS OF NEXT AVAILABLE BYTE
                                        ;       COUNT - UPDATED
                                        ;       R0,R1 DESTROYED
                                        ;
                                        ;-
000706  116767  000000G 000036' STRNXT: MOVB    .PCHAR,CVTBUF           ; MOVE DIGIT TO CONVERTSION BUFFER
000714  012701  000017                  MOV     #17,R1                  ; ASSUME CHAR IS AN '*'
000720  126727  000000G 000052          CMPB    .PCHAR,#'*              ; IS CHAR AN '*' ?
000726  001404                          BEQ     10$                     ; BR IF YES
000730  012700  000036'                 MOV     #CVTBUF,R0              ; POINT TO CONVERSION BUFFER
000734                                  CALL    $CDTB                   ; CONVERT TO BINARY
000740  110177  000040'         10$:    MOVB    R1,@NEXT                ; STORE BINARY DIGIT
000744  005267  000040'                 INC     NEXT                    ; POINT TO NEXT AVAILABLE BYTE
000750  105267  000035'                 INCB    COUNT                   ; INCREMENT DIGIT COUNT
000754                                  RETURN
```

```
     99                                                    .SBTTL  LOCAL SYMBOL DEFINITIONS
    100
    101                                             ;
    102                                             ; LOCAL SYMBOL DEFINITIONS
    103                                             ;
    104        000040                                      SPACE = 40               ; ASCII SPACE
    105        000006                                      PVNAMX = 6               ; Maximum length of circuit id [old format]     ;[TD001]
    106        000020                                      MAXID = 16.              ; Max length of net. mgmt. 'id-string'          ;[TD001]
    107
    108                                             ;
    109                                             ; SAVED REGISTER OFFSETS ON STACK FOR $SAVAL
    110                                             ;
    111        000002                                      R$R0 = 2                 ; SAVED R0
    112        000004                                      R$R1 = 4                 ; SAVED R1
    113        000006                                      R$R2 = 6                 ; SAVED R2
    114        000010                                      R$R3 = 10                ; SAVED R3
    115        000012                                      R$R4 = 12                ; SAVED R4
    116        000014                                      R$R5 = 14                ; SAVED R5
```

```
                                              .SBTTL  BIASX - BIAS XPOOL ADDRESS FOR APR6

                                      ;+
                                      ; BIASX - BIAS XPOOL ADDRESS FOR APR6
                                      ;
                                      ; INPUTS:
                                      ;       R0 = ADDRESS (POOL OR APR5)
                                      ;
                                      ; OUTPUTS:
                                      ;       R0 = ADDRESS (POOL OR APR6)
                                      ;
                                      ;-

001466                                BIASX:
001466  020027  120000                        CMP     R0,#120000      ; POOL ADDRESS?
001472  103404                                BLO     10$             ; YES, LEAVE AS IS
001474  042700  160000                        BIC     #160000,R0      ; CLEAR APR BITS
001500  052700  140000                        BIS     #140000,R0      ; BIAS FOR APR6
001504                                10$:    RETURN                  ; RETURN
```

```
                                                                  K 15
VFGPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:30       PAGE 7
MACRO CROSS REFERENCE                                      CREF    04.00

MACRO NAME          REFERENCES

BIAS                #5-94
CALL                 9-197     11-306    12-350    12-362    12-378    13-409    13-411    13-413    13-414    14-442
                     14-446    14-451    14-452    14-454    14-456    14-457    14-463    14-470    15-494    15-497
                     15-498    15-501    16-536    16-541    16-547    16-564    17-586    17-587    17-598    17-609
                     19-647    19-652    20-731    20-736    20-749
CEACC$              #5-60      15-498    16-536    17-586
CTRDF$              #5-61      5-66
DBGTP$              #10-212   #10-240   #10-244   #10-248   #10-252   #10-256   #10-260   #10-264   #10-268   #10-272
                    #10-276
DTEDF$              #5-61      5-65
EMSG$R              #5-59      9-206     11-316    11-332    11-344    12-357    12-369    12-384    12-398    13-421
                     19-660
GETAD               #5-59      14-452    14-463    15-501    16-541    17-587    17-598    20-736
GETRV               #5-59      14-446    15-494    15-497    20-731
ISTAT$              #5-60      10-212
MAP                 #5-88      20-732
MTRAN$              #10-212
NTLER$              #5-60      8-165     8-166     8-167     8-168     8-169     8-170     8-171     8-172
PHBDF$              #5-60      5-68
PLBDF$              #5-60      5-72
PUTAD               #5-59      14-454    14-470    16-564    17-609    20-749
PVCDF$              #5-59      5-69
RAND                #14-446    14-446   #14-452    14-452   #14-454    14-454   #14-463    14-463   #14-470    14-470
                    #15-494    15-494   #15-497    15-497   #15-501    15-501   #16-541    16-541   #16-564    16-564
                    #17-587    17-587   #17-598    17-598   #17-609    17-609   #20-731    20-731   #20-736    20-736
                    #20-749    20-749
RESMAP              #5-82      13-416    14-455    15-515    17-610    20-750
RESRG               #5-59      15-516
RETURN               9-201     11-312    11-328    11-340    12-353    12-365    12-380    12-394    13-417    14-471
                     15-517    16-565    17-611    18-631    19-656    19-676    19-684    20-751
SAVMAP              #5-78      13-410    14-447    15-491    17-585    20-730
SAVRG               #5-59      15-490
SLTDF$              #5-61      5-70
SOB                  20-725    20-742
STATE$              #5-60      10-217   #10-220   #10-223   #10-226   #10-229   #10-232   #10-235   #10-238   #10-242
                    #10-246   #10-250   #10-254   #10-258   #10-262   #10-266   #10-270   #10-274   #10-280   #10-283
                    #10-286   #10-289   #10-292   #10-295
TRAN$               #5-60     #10-218   #10-221   #10-224   #10-227   #10-230   #10-233   #10-236   #10-239   #10-240
                    #10-243   #10-244   #10-247   #10-248   #10-251   #10-252   #10-255   #10-256   #10-259   #10-260
                    #10-263   #10-264   #10-267   #10-268   #10-271   #10-272   #10-275   #10-276   #10-281   #10-284
                    #10-287   #10-290   #10-293
XCBDF$              #5-59      5-71
XPDDB$              #5-61      5-67
```

```
                                   ;
                                   ; OWNER
                                   ;
000550  022767  000003  000000G  SVOWNR:  CMP       #3,.PSTCN                  ; VALID OWNER?
000556  001402                            BEQ       5$                         ; BR IF YES
000560  000167  000406                    JMP       101$                       ; BR IF NO
000564  016700  000000G          5$:      MOV       .PSTPT,R0                  ; POINT TO OWNER
000570  005001                            CLR       R1                         ;
000572                                    CALL      $CAT5                      ; CONVERT TO RAD50
000576  020127  114224                    CMP       R1,#^RXPT                  ; IS OWNER XPT?
000602  001173                            BNE       101$                       ; BR IF NO - ERROR
000604                                    CALL      FNDOWN                     ; LOOK FOR XPT
000610  103570                            BCS       101$                       ; BR IF XPT NOT FOUND
000612  016767  000000G 000016'           MOV       $SLTA,SLTA                 ; MAKE LOCAL COPY OF $SLTA
000620  016700  000014'                   MOV       STASVC,R0                  ; GET STATION NUMBER
000624  006300                            ASL       R0                         ; MAKE IT A WORD INDEX
000626  016067  000000G 000002'           MOV       $$HTIM(R0),HTIMR           ; STORE HELLO TIMER
000634  016067  000000G 000004'           MOV       $$LTIM(R0),LTIMR           ; STORE LISTEN TIMER
000642  016700  000000G                   MOV       $PSIPT,R0                  ; GET PSI HOME BLOCK ADDRESS
000646  001554                            BEQ       102$                       ; NO PSI HOME BLOCK????

000650                                    CALL      $SAVAL                     ; SAVE ALL REGISTERS
000654  010004                            MOV       R0,R4                      ; SAVE PSI HOME BLOCK ADDRESS
000656  012701  000032                    MOV       #S$LEN,R1                  ; GET LENGTH OF SVC DESCRIPTOR BLOCK
000662                                    CALL      $ALPOL                     ; ALLOCATE SVC DESCRIPTOR BLOCK
000666  103547                            BCS       103$                       ; BR IF UNSUCCESSFUL

000670                           10$:     SAVMAP                               ; SAVE CURRENT MAPPING
000674  010005                            MOV       R0,R5                      ; SAVE CURRENT MAPPING
000676                                    CEACC$    R0                         ; CONVERT TO MAPPED ADDRESS
000706  010046                            MOV       R0,-(SP)                   ; SAVE VIRTUAL ADDRESS OF BLOCK
000710  016746  000000G                   MOV       $RBLCK,-(SP)               ; AND APR BIAS (FOR PLB STORAGE)
000714  012702  000000G                   MOV       #$BFR,R2                   ; POINT TO SVC DESCRIPTOR
000720  105022                   15$:     CLRB      (R2)+                      ; INITIALIZE BLOCK
000722  005301                            DEC       R1                         ; MORE TO INITIALIZE?
000724  003375                            BGT       15$                        ; BR IF YES
000726  116767  000021' 000002G           MOVB      CTLSVC,S$CTL+$BFR          ; STORE CONTROLLER NUMBER
000734  116767  000014' 000003G           MOVB      STASVC,S$STA+$BFR          ; STORE STATION NUMBER
000742  016767  000006' 000004G           MOV       PKTSVC,S$PKSZ+$BFR         ; STORE PACKET SIZE
000750  016767  000010' 000010G           MOV       FLGSVC,S$FLG+$BFR          ; STORE FLAGS BYTE
000756  116767  000023' 000013G           MOVB      RCLSVC,S$RCL+$BFR          ; STORE RECALL TIMER
000764  016767  000012' 000014G           MOV       RTMSVC,S$TIM+$BFR          ; STORE RECALL TIMER (COUNTDOWN)
000772  016767  000012' 000016G           MOV       RTMSVC,S$TMR+$BFR          ; STORE RECALL TIMER (INITIAL VALUE)
001000  116767  000022' 000006G           MOVB      WNDSVC,S$WND+$BFR          ; STORE WINDOW SIZE
001006  116767  000024' 000007G           MOVB      COUNT,S$ADSZ+$BFR          ; STORE SIZE OF DTE ADDRESS (DIGITS)
001014  012702  000010                    MOV       #<DTELEN+1>/2,R2           ; GET NUMBER OF BYTES IN DTE ADDRESS
001020  012701  000020G                   MOV       #$BFR+S$DTE,R1             ; POINT TO DTE ADDRESS
001024  012703  000044'                   MOV       #DTEPCK,R3                 ; GET DTE ADDRESS
001030  112321                   17$:     MOVB      (R3)+,(R1)+                ; STORE DTE ADDRESS
001032  005302                            DEC       R2                         ; MORE TO STORE?
001034  003375                            BGT       17$                        ; BR IF YES
001036  116767  000020' 000030G           MOVB      OWNSVC,S$OWNR+$BFR         ; STORE OWNER
001044                                    PUTAD     R0,#S$LEN                  ; WRITE SVC TO SYSTEM IMAGE FILE
001062  062704  000036                    ADD       #H$SVC,R4                  ; POINT TO SVC LISTHEAD
001066                                    GETRV     R4,#2                      ; READ IN LISTHEAD
001104  010401                   20$:     MOV       R4,R1                      ; SAVE ADDRESS OF PREVIOUS BLOCK
001106  016704  000000G                   MOV       $BFR,R4                    ; GET ADDRESS OF NEXT BLOCK IN LIST
```

```
                                      .SBTTL  SETEVT - SET THE EVENT FILTER MASKS

                              ;+
                              ;
                              ; SETEVT - SET THE EVENT FILTER MASKS
                              ;
                              ; INPUTS:
                              ;       R3 - ADDRESS OF EVENT FILTER MASKS TO BE SET
                              ;       R5 - ADDRESS OF EVENT FILTER MASKS TO OPERATE ON
                              ;
                              ; OUPUTS:
                              ;       R3,R5 DESTROYED
                              ;
                              ;-

 017322  052325               SETEVT: BIS     (R3)+,(R5)+             ; SET THE SPECIFIED EVENT MASKS
 017324  052325                       BIS     (R3)+,(R5)+             ; ...
 017326  052325                       BIS     (R3)+,(R5)+             ; ...
 017330  052325                       BIS     (R3)+,(R5)+             ; ...
 017332                               RETURN
```

```
                                                .SBTTL  X5DSMT - DISMOUNT THE X25ACP
                                        ;+
                                        ; *** - X5DSMT - DISMOUNT THE X25ACP
                                        ;
                                        ; INPUTS:
                                        ;       R5 = NTINIT'S TCB ADDRESS
                                        ;
                                        ; OUTPUTS:
                                        ;       NONE.
                                        ;
                                        ;-

020536                                  X5DSMT:
020536                                          GETRV   R5,.TLGTH               ; READ IN NTINIT'S TCB
020554  016700  000012G                         MOV     T.RCVL+$BFR,R0          ; POINT AT THE SEND DATA PACKET
020560  001415                                  BEQ     10$                     ; IF EQ, ACP NOT MOUNTED
020562                                          GETRV   T.RCVL+$BFR,#36.        ; READ IN PACKET
020602  042767  000400  000006G                 BIC     #LS.CX5,LR.STS+6+$BFR   ; ELSE CLEAR SET X25-S STATE ON
020610                                          PUTRC                           ; REWRITE PACKET
020614                                  10$:    RETURN
```

```
VENA - VNP ENABLE/DISABLE LINES MACRO V05.03b Wednesday 11-Sep-85 12:06  Page 85-3
Symbol table

L.NSTA=  ****** GX     NF.EVT=  000002        N$$BUF=  000001        OP$OUT=  002000        PR$LCC=  000010
L.OWNR=  ****** GX     NF.LLI=  000004        N$$LDV=  000001        OP$OWN=  000020        PR$MOP=  000004
L.PAIC   000017        NF.LV2=  000100        N$$MCP=  000001        OP$PAR=  000010        PR$UP =  000001
L.PAIR   000016        NF.SLI=  000400        N$$MLL=  000001        OP$PRI=  000010        PS$CHR=  000016
L.PLD    000047        NF.SMC=  000020        N$$MOV=  000010        OP$RET=  010000        PS$FAI=  000014
L.PLL    000012        NF.SWP=  000010        N$$NCT=  000001        OP$RPA=  000010        PS$NTI=  000006
L.SCFW   000056        NTINAM   000010R   002 N$$PEM=  000001        OP$SCO=  000001        PS$OFF=  000000
L.STA    000063        NTI11S   000014R   002 OF$INS=  000001        OP$SEG=  000400        PS$STR=  000002
L.STBL   000070        N$ACQ    000000        OF$LOG=  000100        OP$SER=  000400        PS$UP =  000012
L.STRC   000050        N$ACTL   000032        OF$OFF=  000002        OP$SIN=  000010        PS$VER=  000010
L.STRM   000051        N$ADJ1   000072        OF$ON =  000000        OP$STA=  000004        PS$WT =  000004
L.STS    000002        N$ADJ2   000074        OF$SMC=  000200        OP$TER=  000001        PT$BRO=  000200
L.ST2    000003        N$CACH   000062        OF.PRO=  000040        OP$TPA=  000020        PT$DRT=  000100
L.TBP    000004        N$CIR    000034        OF.RLU=  000100        OP$TRI=  000040        PT$END=  000004
L.TIMC   000060        N$CRC    000120        OF.SMC=  000200        OP$UCS=  000020        PT$LV1=  000002
L.TIMI   000001        N$DLA    000020        OPT1     003656R       OP$USE=  000002        PT$LV2=  000001
L.TIMR   000000        N$DLY    000014        OP$ADD=  000200        OP$VEC=  000040        PT$PH3=  000010
L.TOR    000064        N$ELEN   000054        OP$ALL=  000200        OP$VER=  000040        PT$XAR=  000020
L.XSET   000020        N$ENC    000042        OP$CCS=  000004        OP$XAC=  000001        PVALID   000240R   002
MONNAM   000154R   002 N$ERRC   000022        OP$CIE=  004000        OP$XPR=  000002        PX$BLK=  000040
MOVE     011720R       N$FLG    000005        OP$CON=  000400        OP$X5S=  000004        PX$DLM=  000200
MO$SAC=  000016        N$FNC    000006        OP$COP=  000001        OP$X9S=  000010        PX$SVC=  000100
MO$SPR=  000012        N$GENQ   000052        OP$COS=  000100        O.FLG    000003        P$CHN    000004
MO$SSV=  000014        N$GTM    000015        OP$DST=  000200        O.LEN    000012        P$CNT    000005
MO$25A=  000006        N$HC1    000052        OP$DTE=  000040        O.LNK    000000        P$CTR    000034
MO$25P=  000002        N$HC2    000056        OP$DUP=  000002        O.MXC    000004        P$DRCT   000015
MO$25S=  000004        N$HIGH   000033        OP$EST=  000100        O.NAM    000006        P$DRTR   000036
MO$29S=  000010        N$LLT    000026        OP$EVE=  010000        O.TYP    000002        P$FLG    000012
MXCON =  000006        N$LLTM   000024        OP$FIL=  000200        PDVX     000072R   002 P$FWD    000030
MXFIL =  000050        N$LVC    000036        OP$GRP=  000100        PF$BLK=  020000        P$ICCB   000046
MXFILN=  000003        N$LV1    000002        OP$HOS=  000001        PF$CLC=  010000        P$IPL    000014
MXMON =  000006        N$LV2    000010        OP$HTM=  000002        PF$DLM=  100000        P$LCD    000002
MXUNT =  000377        N$MBXQ   000050        OP$INA=  004000        PF$EIP=  000002        P$LEN    000052
MXVER =  077777        N$MHC1   000036        OP$INC=  001000        PF$ENB=  000001        P$LST    000000
M$HIGH=  000003        N$MHC2   000044        OP$INI=  000000        PF$FAI=  004000        P$NRNI   000040
M$$CRB=  000124        N$PLD    000016        OP$KDS=  002000        PF$FM1=  000100        P$OCCB   000050
M$$CRX=  000000        N$PLLT   000030        OP$KDT=  000400        PF$FM2=  000200        P$PFQ    000006
M$$FCS=  000000        N$PRI    000076        OP$KEV=  001000        PF$OFF=  000000        P$PKSZ   000044
M$$MGE=  000000        N$ROA1   000022        OP$KGR=  001000        PF$RM1=  000020        P$RMX1   000016
M$$NET=  000000        N$ROA2   000030        OP$KNT=  010000        PF$RM2=  000040        P$RMX2   000020
M$$OVR=  000000        N$RTMX   000014        OP$KSK=  002000        PF$RVR=  000010        P$RPRI   000042
M$3100=  000000        N$RTM1   000014        OP$LCT=  000200        PF$STA=  000004        P$RTIM   000003
M$3101=  000001        N$RTM2   000015        OP$LNE=  000040        PF$SVC=  040000        P$STA1   000022
M$3102=  000002        N$RT1    000000        OP$LOC=  000001        PF$UP =  000004        P$STA2   000023
M$3103=  000003        N$RT2    000006        OP$LST=  000002        PH$HDE=  000004        P$TIM    000010
NF$ACC=  100000        N$SLA    000016        OP$LTM=  000001        PH$LOC=  000002        P$TSCT   000026
NF$BLK=  000100        N$SNOD   000012        OP$LVL=  000010        PH$MTS=  000003        P$TSIZ   000024
NF$DMO=  000010        N$TCTL   000112        OP$MAC=  000010        PH$UMP=  000000        P$TYP    000001
NF$EVT=  040000        N$TIM    000004        OP$MCO=  000002        PH$WCS=  000001        P$$P45=  000000
NF$MOU=  000040        N$TLC    000100        OP$MDE=  000400        PLBFLG   000122R   002 P$$WRD=  000000
NF$RST=  000002        N$TNC    000114        OP$MLI=  000004        PROEVT   012422R       P.ACT    000012
NF$SCN=  000020        N$TRC    000106        OP$MON=  000100        PRVUNM   000166R   002 P.CFN    000034
NF$SHU=  000004        N$TTCB   000110        OP$MST=  000020        PR$BED=  000200        P.CHAR   000042
NF$TIM=  000200        N$VCB    000010        OP$NAM=  000004        PR$BEU=  000100        P.CONT   000067
NF.ACC=  000001        N$VER    000066        OP$NET=  004000        PR$BRD=  000040        P.DEDT   000100
NF.BCP=  000040        N$XLEN   000124        OP$NLI=  000002        PR$BRU=  000020        P.DELT   000076
NF.END=  000200        N$$ACC=  000001        OP$NOD=  000020        PR$DWN=  000002        P.DISL   000006
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08     PAGE 10
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE            REFERENCES

L.COST   = ******  GX    *17-783
L.DLC    = ******  GX     20-907    21-965    22-1019
L.DLM    = ******  GX     20-918    21-976    22-1030
L.DLS    = ******  GX     20-912    21-970    22-1024
L.FLG    = ******  GX     11-451    11-454    11-484    11-496   *12-542   *12-550   *12-568    13-592    14-634
                          15-666    16-704    16-706    16-713    17-773    17-776    17-784    20-905    21-963
                          22-1017  *23-1067  *23-1075  *24-1105  *24-1124  *24-1131  *24-1148  *33-1742  *34-1770
                          81-4252
L.MPF    = ******  GX     11-468    12-555    16-732    17-796    24-1114   24-1138   33-1747   34-1775   81-4258
L.NMST   = ******  GX    *23-1068  *24-1106
L.NSTA   = ******  GX     11-459    11-482    11-498    12-543    12-548    16-708    17-778    23-1069   23-1076
                          24-1100   24-1129   33-1740   34-1768   81-4255
L.OWNR   = ******  GX    *16-716    81-4254
L.PLD      000047        *15-672
L.PLL      000012         20-919    20-923
MONNAM     000154 R      #6-157     48-2729  *58-3270  *58-3271   58-3273   58-3274
MOVE       011720 R       42-2373   42-2397   42-2401   42-2405   42-2409   42-2413  #42-2445
MO$SAC   = 000016        #5-79
MO$SPR   = 000012        #5-79
MO$SSV   = 000014        #5-79
MO$25A   = 000006        #5-79
MO$25P   = 000002        #5-79
MO$25S   = 000004        #5-79
MO$29S   = 000010        #5-79
MXCON    = 000006        #8-205     47-2692   49-2766
MXFIL    = 000050        #8-207     47-2692   50-2799
MXFILN   = 000003        #8-210     50-2866
MXMON    = 000006        #8-206     47-2692   48-2730
MXUNT    = 000377        #8-209     52-2990
MXVER    = 077777        #8-208     50-2913
M$HIGH   = 000003        #5-79
M$$MGE   = 000000         5-82      5-82
M$3100   = 000000        #5-79
M$3101   = 000001        #5-79
M$3102   = 000002        #5-79
M$3103   = 000003        #5-79      5-79
NF$ACC   = 100000         38-1915   38-1918
NF$MOU   = 000040         11-443
NF.ACC   = 000001         38-1913
NTINAM     000010 R      #6-120     40-2107   75-4089
NTI11S     000014 R      #6-121     39-1965   40-2106
N$ELEN     000054         11-442
N$FLG      000005         11-443
N$PLD      000016        *74-4029  *74-4030   75-4104   75-4108   75-4111  *75-4113  *75-4114
N$TLC      000100         81-4272   81-4273   81-4274
N$VER      000066         36-1831   37-1871
N$XLEN     000124         36-1830   37-1870   74-4028   75-4103   81-4271
N$$VCT   = ******         28-1569   29-1619   30-1648   32-1712   41-2272   42-2422   45-2592   59-3340   60-3378
                          61-3432   62-3502   62-3514   62-3537   62-3577   62-3586   62-3615   62-3635   63-3676

OF$INS   = 000001        #5-73
OF$LOG   = 000100        #5-73      41-2241
OF$OFF   = 000002        #5-73
```

```
VENA        CREATED BY  MACRO  ON 11-SEP-85 AT 12:08       PAGE 23

MACRO CROSS REFERENCE                                      CREF   04.00

MACRO NAME      REFERENCES

                #60-3384    60-3384    #61-3433     61-3433    #61-3438     61-3438    #62-3503     62-3503    #62-3515     62-3515
                #62-3540    62-3540    #62-3580     62-3580    #62-3589     62-3589    #62-3616     62-3616    #62-3638     62-3638
                #63-3677    63-3677    #63-3682     63-3682    #74-4009     74-4009    #74-4013     74-4013    #74-4025     74-4025
                #74-4028    74-4028    #74-4031     74-4031    #74-4065     74-4065    #75-4094     75-4094    #75-4097     75-4097
                #75-4103    75-4103    #75-4115     75-4115    #75-4123     75-4123    #76-4139     76-4139    #76-4142     76-4142
                #76-4144    76-4144    #77-4164     77-4164    #78-4179     78-4179    #78-4188     78-4188    #79-4208     79-4208
                #80-4228    80-4228    #81-4271     81-4271    #81-4275     81-4275    #82-4321     82-4321    #82-4328     82-4328
                #82-4333    82-4333    #82-4344     82-4344    #82-4346     82-4346    #82-4348     82-4348    #84-4415     84-4415
                #84-4422    84-4422
RESRG           #5-62       32-1721     62-3504     62-3516     62-3541     62-3581     62-3590     62-3617     62-3639     63-3683
RETURN            11-511    12-571      13-610      14-642      15-674      16-748      17-798      18-843      19-878      20-935
                  21-991    22-1044     23-1082     24-1149     25-1213     26-1276     26-1294     26-1311     26-1328     26-1345
                  26-1362   27-1461     28-1586     29-1625     30-1654     31-1678     31-1686     32-1722     33-1750     34-1778
                  35-1814   36-1854     37-1892     38-1920     39-2054     39-2067     40-2088     40-2094     40-2162     40-2168
                  40-2181   41-2279     42-2430     42-2450     43-2475     44-2502     44-2530     45-2569     45-2590     46-2659
                  47-2708   48-2750     49-2782     50-2920     51-2953     52-3001     53-3038     54-3069     55-3107     56-3153
                  57-3199   57-3224     58-3289     59-3348     60-3403     61-3460     62-3645     63-3709     64-3730     65-3755
                  66-3785   67-3807     68-3833     69-3854     70-3874     71-3914     72-3942     73-3981     73-3985     74-4074
                  75-4124   76-4145     77-4168     78-4190     79-4209     80-4229     81-4285     82-4349     83-4381     83-4398
                  84-4427   85-4459
RNBDF$          #5-60       5-74
SAVRG           #5-62       32-1707     62-3501     62-3513     62-3536     62-3576     62-3585     62-3614     62-3634     63-3675
SWTCHI          #5-61       27-1398     27-1400     28-1572     28-1574     42-2338     42-2340     42-2389     42-2391     45-2584
                  53-3019   53-3022     63-3672     63-3708
SWTCHO          #5-60       27-1406     27-1408     28-1582     28-1584     39-2020     40-2161     42-2344     42-2346     42-2419
                  42-2426   42-2428     45-2583     62-3520     62-3544     62-3575     62-3620     62-3643     82-4347
TCBDF$          #5-59       5-66
TMPDF$          #5-82       5-82
UCBDF$          #5-61       5-78
VNPBPT          #11-395     11-396
VNPDF$          #5-60       5-73
XPDDB$          #5-59       5-71
.ADDRB          #5-82
.ADDRW          #5-82
.APR            #5-82
.BIN            #5-82
.CNB            #5-82
.CNW            #5-82
.CORE           #5-82
.CSR            #5-82       5-82        5-82
.CTIM           #5-82
.DPRB           #5-82
.DPRW           #5-82
.DVCHA          #5-82       5-82
.INT            #5-82       5-82        5-82
.INT1           #5-82
.INT2           #5-82
.INT3           #5-82
.LFLHD          #5-82
.LIBR           #5-82
.LINKS          #5-82
.LSTHD          #5-82       5-82
```

```
                                  ;+
                                  ; $VEX2 - EXPAND BINARY INFORMATION INTO REAL INFORMATION
                                  ;
                                  ; THIS ROUTINE IS CALLED AFTER THE ASCII TEMPLATE FILE HAS BEEN READ
                                  ; AND PROCESSED INTO THE BINARY FILE. THE BINARY INFORMATION IS EXPANDED
                                  ; AND STORED AT THE LOCATION CONTAINED IN "$ADDR". "$LINK" CONTAINS A
                                  ; BASE ADDRESS FOR THE REAL DATA WHICH NEED NOT BE THE SAME AS "$ADDR".
                                  ;
                                  ; INPUTS:
                                  ;       $BIN=START OF BINARY INFORMATION
                                  ;       $PTR=END OF BINARY INFORMATION
                                  ;       $ADDR=ADDRESS OF WHERE REAL DATA GOES
                                  ;       $LINK=LINK VALUE FOR REAL DATA
                                  ;
                                  ; OUTPUTS:
                                  ;       ALL REGISTERS=DESTROYED
                                  ;-
000000                            $VEX2::
000000  005067  000446'                   CLR     INDEX           ; ZERO INDEX VALUE
000004  005067  000450'                   CLR     FLAGS           ; AND SPECIAL FLAGS
000010                                    CALL    PRLT            ; DO WE HAVE A PARTITION LINE TABLE?
000014  016705  000000G                   MOV     $BIN,R5         ; START OF BINARY DATA
000020  016704  000000G                   MOV     $ADDR,R4        ; START OF REAL DATA
000024  026705  000000G           10$:    CMP     $PTR,R5         ; IS IT END OF BINARY DATA?
000030  001411                            BEQ     20$             ; IF EQ, YES
000032  005003                            CLR     R3              ; GET FUNCTION CODE
000034  152503                            BISB    (R5)+,R3        ;       ...
000036  001406                            BEQ     20$             ; IF ZERO, END OF FILE
000040  106303                            ASLB    R3              ; CONVERT TO A WORD INDEX
000042                                    CALL    @TABLE-2(R3)    ; EXPAND ONE FIELD
000046                                    CALL    $BUFCK          ; DON'T LET BUFFER OVERFILL
000052  000764                            BR      10$             ; KEEP LOOPING UNTIL END
000054                            20$:    CALL    WRVBL           ; WRITE THE ICB
000060                                    RETURN
```

```
                                        ;
                                        ; SECONDARY MUX FILE ADDRESS
                                        ;
001754  012767  100000  000450' MXPTB:  MOV     #100000,FLAGS       ; INDICATE MUX EXPANSION
001762  016746  000000G                 MOV     $LINK,-(SP)         ; SAVE PRIMARY TEMPLATE RELOCATION BIAS
001766  016767  000000G 000000G         MOV     $MXADD,$LINK        ; GET MUX TEMPLATE ADDRESS
001774  160467  000000G                 SUB     R4,$LINK            ; COMPUTE RELOCATION BIAS FROM BEGINNING
002000  162767  000002  000000G         SUB     #2,$LINK            ; OF MUX TEMPLATE EXPANSION
002006  112546                          MOVB    (R5)+,-(SP)         ; GET REPEAT COUNT
002010  112524                          MOVB    (R5)+,(R4)+         ; GET MUX TABLE ADDRESS
002012  112524                          MOVB    (R5)+,(R4)+         ; ...
002014  010446                          MOV     R4,-(SP)            ; SAVE CURRENT BINARY FILE POINTER
002016  010546                          MOV     R5,-(SP)            ; SAVE CURRENT BINARY FILE POINTER

002020  011605                  10$:    MOV     (SP),R5             ; RESTORE BINARY FILE POINTER
002022  005003                  20$:    CLR     R3                  ; GET FUNCTION CODE
002024  152503                          BISB    (R5)+,R3            ; ...
002026  122703  000000G                 CMPB    #.EOF,R3            ; IS IT END OF FILE ?
002032  001404                          BEQ     30$                 ; IF EQ, YES
002034  106303                          ASLB    R3                  ; CONVERT TO A WORD INDEX
002036                                  CALL    @TABLE-2(R3)        ; EXPAND A FIELD
002042  000767                          BR      20$                 ; GO TILL END OF FILE
002044  005267  000446'         30$:    INC     INDEX               ; TALLY ONE PASS THRU SECONDARY FILE
002050  126667  000004  000446'         CMPB    4(SP),INDEX         ; ARE WE DONE ?
002056  001360                          BNE     10$                 ; IF NE, NO
002060  005726                          TST     (SP)+               ; ELSE, PURGE BINARY FILE POINTER
002062  010402                          MOV     R4,R2               ; SAVE CURRENT POSITION IN OUTPUT FILE
002064  012604                          MOV     (SP)+,R4            ; RESTORE OUTPUT FILE POINTER
002066  005726                          TST     (SP)+               ; PURGE COUNT
002070  160402                          SUB     R4,R2               ; CALCULATE SIZE OF MUX TABLES

002072                                  SAVRG   <R4>                ; WILL NEED R4 LATER FOR REST OF LINE TABLE
002074  016400  177776                  MOV     -2(R4),R0           ; R0 WILL BE POINTER FOR IMAGE
002100  010203                          MOV     R2,R3               ; R3 WILL BE REMAINING BYTE COUNT
002102  022702  001000          35$:    CMP     #1000,R2            ; IS BYTE COUNT GREATER THAN LIMIT ?
002106  002002                          BGE     40$                 ; IF GE, NO ... JUST MOVE THAT MANY BYTES
002110  012702  001000                  MOV     #1000,R2            ; ELSE MOVE MAX ALLOWED
002114                          40$:    SWTCHO  R4                  ; POINT OUTPUT BUFFER TO PROPER LOCATION
002120                                  PUTRC   R0,R2               ; WRITE TO IMAGE
002134  062700  001000                  ADD     #1000,R0            ; UPDATE IMAGE POINTER
002140  062704  001000                  ADD     #1000,R4            ; UPDATE OUTPUT FILE POINTER
002144  160203                          SUB     R2,R3               ; UPDATE BYTE COUNT
002146  010302                          MOV     R3,R2               ; SET UP FOR NEXT PASS THRU LOOP
002150  001354                          BNE     35$                 ; IF NE, MORE BYTES REMAIN TO BE MOVED
002152                                  RESRG   <R4>                ; RESTORE OUTPUT FILE POINTER
002154                                  SWTCHO                      ; RESET OUTPUT BUFFER

002162  005067  000446'                 CLR     INDEX               ; CLEAR SECONDARY INDEX COUNT
002166  005067  000450'                 CLR     FLAGS               ; AND SPECIAL FLAGS
002172  016703  000000G                 MOV     $MXSIZ,R3           ; GET SIZE OF MUX TABLES
002176  012667  000000G                 MOV     (SP)+,$LINK         ; RESTORE PRIMARY TEMPLATE RELOCATION BIAS
002202                                  CALLR   SAVALC              ; SAVE ADDRESS AND SIZE IN UNLOAD BLOCK

                                        ;
                                        ; CURRENT TIME (TIME)
                                        ;
002206  005024                  TIME:   CLR     (R4)+               ; STORE TIME (SAY IT'S MIDNIGHT)
```

```
        .TITLE  VEX3 - VNP EXPANSION ROUTINES CLEANUP
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       VNP - ERROR CASE LINE TABLE RESOURCE DE-ALLOCATOR
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;



```

```
		.TITLE	VEX5 - VNP UPDATE LINE TABLES
		.IDENT	/V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;	VNP - MUX CHARACTERISTICS UPDATE LINE TABLE SCAN
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	27-FEB-78
;	VERSION 2.0 RELEASE
;
; 2.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;



```

```
    668 001642  011267  000162'                    MOV     (R2),CTRADD       ; SAVE CONTROLLER TABLE STARTING ADDRESS
    669 001646  010267  000160'                    MOV     R2,UBA            ; SAVE UNLOAD ENTRY ADDRESS WITHIN BLOCK
    670 001652  017767  000000G 000156'            MOV     @$ADRX,UBIAS      ;  AND ADDRESS
    671 001660  012702  000164'                    MOV     #BLOCK,R2         ; RESET R2 TO START OF BLOCK
    672
    673 001664                           100$:     SWTCHI                    ; POINT BACK AT DEFAULT BUFFER
    674 001672                                     RETURN
```

```
     99                                         .SBTTL  LOCAL DATA
    100
    101                                 ;
    102                                 ; LOCAL DATA
    103                                 ;
    104
    105 000000                                  .PSECT  DATA,D
    106
    107                                         .NLIST  BEX
    108
    109                                 ;
    110                                 ; LOCAL DATA FOR DTE$DF MACRO
    111                                 ;
    112                 000017                  DTEALN = 15.                    ; MAXIMUM NUMBER OF DIGITS IN DTE ADDRESS
    113 000000                          DTEADD: .BLKB   DTEALN                  ; TEMPORARY STORAGE FOR UNPACKED DTE ADDRESS
    114                 000016'                 DTEND = .-1
    115 000017                          DTEPCK: .BLKB   8.                      ; PACKED DTE ADDRESS
    116                 000006                  DTNMMX = 6.                     ; MAX CHARACTERS IN NETWORK NAME
    117 000027                          DTNTNM: .BLKB   DTNMMX                  ; NETWORK NAME
    118 000035                          COUNT:  .BLKB   1                       ; COUNT OF NUMBER OF DIGITS IN DTE ADDRESS
    119 000036                          CVTBUF: .BLKB   1                       ; CONVERSION BUFFER FOR ASCII TO DECIMAL DIGIT
    120                                         .EVEN
    121 000040                          NEXT:   .BLKW   1                       ; ADDRESS OF NEXT AVAILABLE BYTE
    122 000042                          LINNAM: .BLKW   1                       ; DEVICE NAME FROM LINE-ID (RAD50)
    123 000044                          LINCTL: .BLKB   1                       ; CONTROLLER NUMBER
    124 000045                          LINUNT: .BLKB   1                       ; UNIT NUMBER
```

```
                                                .SBTTL  PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
                                        ;+
                                        ;
                                        ; PCKBCD - PACK A STRING OF DIGITS IN BCD FORMAT
                                        ;
                                        ; INPUTS:
                                        ;       R0 - NUMBER OF BYTES IN PACKED ADDRESS
                                        ;       R4 - ADDRESS OF BUFFER CONTAINING ASCI DIGITS TO PACK (0=END OF BUFFER)
                                        ;       R5 - ADDRESS OF BUFFER TO STORE PACKED DATA
                                        ;       COUNT - NUMBER OF DIGITS SPECIFIED
                                        ;
                                        ; OUTPUTS:
                                        ;       BUFFER SPECIFIED IN R5 CONTAINS DATA IN BCD FORMAT
                                        ;
                                        ;-

000756  010501                          PCKBCD: MOV     R5,R1                   ; GET ADDRESS OF PACKED DATA
000760  116702  000035'                         MOVB    COUNT,R2                ; GET NUMBER OF DIGITS IN STRING
000764  001423                                  BEQ     40$                     ; BR IF NOTHING TO DO
000766  105021                          5$:     CLRB    (R1)+                   ; INITIALIZE PACKED ADDRESS
000770  005300                                  DEC     R0                      ; MORE TO INITIALIZE?
000772  003375                                  BGT     5$                      ; BR IF YES
000774  005001                                  CLR     R1                      ; POSITION INIDCATOR (0=HIGH FOUR BITS)
000776  005305                                  DEC     R5                      ; UPDATE ADDRESS FOR FIRST TIME
001000  112400                          10$:    MOVB    (R4)+,R0                ; GET DIGIT TO PACK
001002  005701                                  TST     R1                      ; STORE IN HIGH FOUR BITS?
001004  001007                                  BNE     20$                     ; BR IF NO
001006                                          ASL$    4,R0                    ; MOVE TO HIGH FOUR BITS
001016  005201                                  INC     R1                      ; INDICATE STORE IN LOW FOUR BITS NEXT
001020  005205                                  INC     R5                      ; POINT TO NEXT BYTE
001022  000401                                  BR      30$                     ; BR TO STORE DECIMAL
001024  005301                          20$:    DEC     R1                      ; INDICATE STORE IN HIGH FOUR BITS NEXT
001026  150015                          30$:    BISB    R0,(R5)                 ; PACK THE NUMBER INTO THE BUFFER
001030  005302                                  DEC     R2                      ; MORE TO CONVERT?
001032  003362                                  BGT     10$                     ; BR IF YES
001034                                  40$:    RETURN
        000001                                  .END
```

```
    118                                        .SBTTL  LOCAL DATA
    119
    120                                ;
    121                                ; LOCAL  DATA
    122                                ;
    123
    124 000000                                 .PSECT  DATA,D
    125
    126                                        .NLIST  BEX
    127
    128                                ;
    129                                ; LOCAL  DATA FOR PVC$DF MACRO
    130                                ;
    131 000000                         HTIMR:   .BLKW    1                ; LOCAL COPY OF HELLO TIMER
    132 000002                         LTIMR:   .BLKW    1                ; LOCAL COPY OF LISTEN TIMER
    133 000004                         PORTNO:  .BLKW    1                ; NUMBER OF PORTS
    134 000006                         PRTADD:  .BLKW    1                ; ADDRESS OF ENTRY IN PORT TABLE
    135 000010                         PVCLCN:  .BLKW    1                ; LOGICAL CHANNEL NUMBER
    136 000012                         PVCCTM:  .BLKW    1                ; COUNTER TIMER
    137 000014                         PVCPSZ:  .BLKW    1                ; MAXIMUM BLOCK SIZE
    138 000016                         PVCWSZ:  .BLKW    1                ; WINDOW SIZE
    139 000020                         PVCOWN:  .BLKW    1                ; OWNER PROCESS
    140 000022                         DLMPDV:  .BLKW    1                ; ADDRESS OF DLM PDV
    141 000024                         DLMSLT:  .BLKW    1                ; ADDRESS OF DLM SLT
    142 000026                         DLMSTA:  .BLKW    1                ; DLM'S STATION NUMBER
    143 000030                         DLMSLN:  .BLKW    1                ; DLM'S SYSTEM LINE NUMBER
    144 000032                         FLAG:    .BLKW    1                ; LOCAL FLAGS WORD
    145              000001                     PF.DLM = 1                ; PVC IS A DLM CIRCUIT
    146 000034                         PVCPRT:  .BLKB    1                ; PORT NUMBER OF FREE ENTRY
    147 000035                         PVCFLG:  .BLKB    1                ; PVC FLAGS BYTE
    148 000036                         PVCNML:  .BLKB    1                ; Length of circuit identification          ;[TD001]
    149 000037                         PVCNAM:  .BLKB    MAXID            ; Circuit identification                    ;[TD001]
    150 000057                         DLMCTL:  .BLKB    1                ; DLM'S CONTROLLER NUMBER
    151 000060        000              DLMUNT:  .BYTE    0                ; DLM'S UNIT NUMBER (ALWAYS 0, MUST BE MTP)
    152 000061                         DLMIND:  .BLKB    1                ; DLM'S PDV INDEX
    153                                         .EVEN
    154                                ;
    155                                ; STATE VALUE FOR "DATA TRANSFER STATE" FOR XCB
    156                                ; GLOBAL PATCHABLE JUST IN CASE ...
    157                                ;
    158 000062 000003                  SP$P4:: .WORD    3                 ; VALUE FOR DATA TRANSFER STATE
    159
```

```
                                                .SBTTL  PVDLM - CHECK FOR DLM PVC

                                        ;+
                                        ;
                                        ; PVDLM - CHECK FOR DLM PVC
                                        ;
                                        ; INPUTS:
                                        ;       PVCNAM - PROCESS NAME (ASCII)
                                        ;
                                        ; OUTPUTS:
                                        ;       DLMPDV - DLM'S PDV ADDRESS
                                        ;
                                        ;-
001506  012700  000037'                 PVDLM:  MOV     #PVCNAM,R0      ; POINT TO SPECIFIED PROCESS NAME
001512                                          CALL    $CAT5           ; CONVERT TO RAD50
001516  020127  015355                          CMP     R1,#^RDLM       ; IS THIS DLM?
001522  001011                                  BNE     INVCIR          ; BR IF NO - ILLEGAL CIRCUIT ID
001524  010100                                  MOV     R1,R0           ; GET DLM'S RAD50 PROCESS NAME

001526                                          CALL    FNDPDV          ; GET DLM'S PDV INDEX
001532  103405                                  BCS     INVCIR          ; BR IF NOT FOUND
001534  110167  000061'                 10$:    MOVB    R1,DLMIND       ; SAVE PDV INDEX
001540  010067  000022'                         MOV     R0,DLMPDV       ; SAVE DLM'S PDV ADDRESS
001544                                          RETURN
                                        ;
                                        ; ERRORS
                                        ;
001546                                  INVCIR: EMSG$R  NO              ; ILLEGAL CIRCUIT ID
                                        ;
                                        ; PVCTL - GET CONTROLLER NUMBER FOR DLM
                                        ;
                                                .ENABL  LSB
001554  116767  000000G 000057' PVCTL:  MOVB    .PNUMB,DLMCTL   ; SAVE CONTROLLER NUMBER
001562  005767  000000G                         TST     .PNUMH          ; LEGAL VALUE?
001566  001367                                  BNE     INVCIR          ; BR IF NO - ERROR
001570  105767  000001G                         TSTB    .PNUMB+1        ; LEGAL VALUE?
001574  001364                                  BNE     INVCIR          ; BR IF NO
001576  016700  000000G                         MOV     $SLTA,R0        ; GET SLT ADDRESS
001602  126760  000000G 000012                  CMPB    .PNUMB,L.CTL(R0); IS THIS THE CORRECT CONTROLLER NUMBER?
001610  001405                                  BEQ     7$              ; CHECK FOR VALIDITY
001612  005067  000000G                 5$:     CLR     SYNERR          ; ELSE NO SYNTAX ERROR
001616  062716  000002                          ADD     #2,(SP)         ; AND REJECT TRANSITION
001622  000416                                  BR      20$             ; ...
001624                                  7$:     RETURN
001626  116767  000000G 000026' PVSTA:  MOVB    .PNUMB,DLMSTA   ; SAVE STATION NUMBER
001634  126767  000000G 000000G                 CMPB    .PNUMB,$TRIB    ; CORRECT STATION NUMBER?
001642  001363                                  BNE     5$              ; BR IF NO
001644  005767  000000G                 10$:    TST     .PNUMH          ; LEGAL VALUE?
001650  001336                                  BNE     INVCIR          ; BR IF NO - ERROR
001652  105767  000001G                         TSTB    .PNUMB+1        ; LEGAL VALUE?
001656  001333                                  BNE     INVCIR          ; BR IF NO
001660                                  20$:    RETURN
                                                .DSABL  LSB
```

L 15

```
**FILE**ID**VFGSVC




VV      VV   FFFFFFFFFF     GGGGGGGG     SSSSSSSS   VV      VV     CCCCCCCC
VV      VV   FFFFFFFFFF     GGGGGGGG     SSSSSSSS   VV      VV     CCCCCCCC
VV      VV   FF           GG           SS           VV      VV   CC
VV      VV   FF           GG           SS           VV      VV   CC
VV      VV   FF           GG           SS           VV      VV   CC
VV      VV   FF           GG           SS           VV      VV   CC
VV      VV   FFFFFFFF     GG             SSSSSS     VV      VV   CC
VV      VV   FFFFFFFF     GG             SSSSSS     VV      VV   CC
VV      VV   FF           GG  GGGGGG           SS   VV      VV   CC
VV      VV   FF           GG  GGGGGG           SS   VV      VV   CC
  VV  VV     FF           GG      GG           SS     VV  VV     CC               ....
  VV  VV     FF           GG      GG           SS     VV  VV     CC               ....
    VV       FF             GGGGGG     SSSSSSSS         VV         CCCCCCCC       ....
    VV       FF             GGGGGG     SSSSSSSS         VV         CCCCCCCC       ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
    473 001112  001414                  BEQ     30$                     ; BR IF END OF LIST - LINK IT HERE
    474 001114                          CEACC$  R4,R3                   ; CONVERT TO MAPPED ADDRESS
    475 001124                          GETAD   R3,#2                   ; READ IN LISTHEAD
    476 001142  000760                  BR      20$                     ; GET NEXT BLOCK IN LIST
    477 001144  010546          30$:    MOV     R5,-(SP)                ; GET UNMAPPED ADDR OF CURRENT BLOCK
    478 001146  010146                  MOV     R1,-(SP)                ; GET ADDRESS OF PREVIOUS BLOCK
    479 001150                          CALL    $XLINK                  ; LINK BLOCK INTO LIST
    480 001154  012600                  MOV     (SP)+,R0                ; GET APR BIAS OF SVC DESCRIPTOR BLOCK
    481 001156  012601                  MOV     (SP)+,R1                ; GET VIRTUAL ADDRESS OF SVC BLOCK
    482 001160                          CALL    SETTLC                  ; SET UP TRANSPORT LINE COUNTERS
    483 001164                          RESMAP                          ; RESTORE PREVIOUS MAPPING
    484 001170                          RETURN
    485
    486                         ;
    487                         ; ERROR
    488                         ;
    489 001172                  101$:   EMSG$R  3J                      ; ILLEGAL OWNER
    490 001200                  102$:   EMSG$R  3Z                      ; PSI NOT LOADED
    491 001206                  103$:   EMSG$R  3K                      ; ALLOCATION FAILURE
```

```
                                             .SBTTL  CLREVT - CLEAR THE SPECIFIED EVENT MASKS

                                     ;+
                                     ;
                                     ; CLREVT - CLEAR THE SPECIFIED EVENT FILTER MASKS
                                     ;
                                     ; INPUTS:
                                     ;       R3 - ADDRESS OF EVENT FILTER MASKS TO BE CLEARED
                                     ;       R5 - ADDRESS OF EVENT FILTER MASKS TO OPERATE ON
                                     ;
                                     ; OUTPUTS:
                                     ;       CARRY CLEAR:
                                     ;          EVENT MASKS ARE STILL SET IN THE FILTER BLOCK SPECIFIED IN R5
                                     ;       CARRY SET:
                                     ;          ALL EVENT FILTER MASKS ARE CLEAR
                                     ;
                                     ;-

 017334  042325                      CLREVT: BIC     (R3)+,(R5)+             ; CLEAR THE SPECIFIED FILTER MASKS
 017336  042325                              BIC     (R3)+,(R5)+             ; ...
 017340  042325                              BIC     (R3)+,(R5)+             ; ...
 017342  042325                              BIC     (R3)+,(R5)+             ; ...
 017344                                      CALL    CHKEVT                  ; CHECK IF ANY MASKS STILL SET
 017350                                      RETURN
```

```
                                                .SBTTL  FNDDCB - FIND A DCB ADDRESS
                                        ;+
                                        ; *** - FNDDCB - FIND A DCB ADDRESS
                                        ;
                                        ; INPUTS:
                                        ;       R0 = DEVICE NAME TO SEARCH FOR
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;
                                        ;-

020616                                  FNDDCB:
020616  016767  000000G 000000G                 MOV     $DEVHD,$BFR             ; GET DEVICE LIST
020624  016767  000000G 000000G         10$:    MOV     $BFR,$RADDR             ; GET NEXT DEVICE IN LIST
020632  000261                                  SEC                             ; ASSUME FAILURE
020634  001411                                  BEQ     20$                     ; IF EQ, END OF LIST
020636                                          GETRV   ,#D.PCB+2               ; READ IN DCB
020650  020067  000000G                         CMP     R0,$BFR+D.NAM           ; IS THIS THE DCB ?
020654  001363                                  BNE     10$                     ; IF NE, NO
020656  000241                                  CLC                             ; ELSE, INDICATE SUCCESS
020660                                  20$:    RETURN
```

```
P.DISR  000004              RE$OPR= 000000          SERFLG  000126R     002 S.NKLR  000040          S1.RSP= 000004
P.DVRF  000047              RE$RCV= 000001          SETBIT  005774R         S.NKR   000037          S1.RST= 000001
P.DVTM  000046              RE$SED= 000011          SETEVT  017322R         S.NKRB  000052          S1.TNE= 040000
P.ECCB  000064              RE$SKW= 000006          SETFIL  015110R         S.NKRW  000045          S1.TSY= 000040
P.ENBC  000011              RE$STA= 000002          SETQLF  017466R         S.NKSB  000051          S1.USI= 100000
P.HDER  000056              RE$SUM= 000003          SE.IRS= 004000          S.NKSW  000046          S2.BEL= 020000
P.INPQ  000036              RE$SYN= 000000          SE.NRS= 002000          S.NMST= ****** GX       S2.BRQ= 000020
P.LEN = 000154              RE$TME= 000021          SE.RBS= 001000          S.NTD   000024          S2.CR  = 002000
P.LERR  000054              RE$TMO= 000000          SE.RBU= 000400          S.OWNR= ****** GX       S2.CTO= 040000
P.LINE  000010              RE$TMR= 000020          SE.RCH= 000001          S.PLA   000010          S2.CTS= 100000
P.MODE  000060              RE$UPT= 000002          SE.RDC= 000002          S.PLC   000022          S2.ELF= 001000
P.MODI  000061              RE$URE= 000003          SE.RRR= 000004          S.PLL   000006          S2.FLF= 000400
P.MTER  000057              RE$VER= 000005          SE.SBS= 000200          S.PLS   000011          S2.IRQ= 000200
P.OFFS= 000046 G            RE$VRQ= 000015          SE.SBU= 000100          S.PSA   000003          S2.OBF= 004000
P.PARM  000070              RF.CTL= 000003          SE.SDC= 000020          S.RBE   000042          S2.ORQ= 000100
P.PIND  000066              RF.LD1= 000040          SE.SHC= 000010          S.RCB   000020          S2.PCU= 010000
P.POLT  000102              RF.LD2= 000100          SE.SRR= 000040          S.RCV   000014          S2.RCU= 000001
P.RAQC  000020              RF.LOO= 100000          SF.ACT= ****** GX       S.RCVB  000072          S2.SRQ= 000040
P.RASQ  000022              RF.TIM= 177400          SF.ENA= ****** GX       S.RCVC  000056          S2.WAL= 000010
P.RBFC  000021              RF.TMO= 000400          SF.LPB= ****** GX       S.RDE   000034          S2.WRA= 000006
P.RERR  000052              RF.WFC= 000200          SF.SER= ****** GX       S.REPR  000050          S2.WRB= 000002
P.SEL   177776              RF.WTD= 000020          SINKND  000130R     002 S.REPS  000047          S3.ACR= 000001
P.SPFC  000104              RF.WTM= 000030          SNAPX   000124R     002 S.RST   000036          S3.CTC= 000004
P.STAT  000002              RF.WTS= 000010          SNKTYP  000230R     002 S.RTEC  000042          S3.FDX= 000200
P.STMT  000074              R[illegible]    = ****** GX  SNODTB  003660R         S.RTY   000001          S3.ICE= 001000
P.TBAD  000062              [illegible] I= 000002     STQUAL  017650R         S.SELC  000066          S3.MHE= 000400
P.TIMC  000050              [illegible] F= 000001     STREVT  017432R         S.SELT  000053          S3.NEC= 000020
P.TIME  000001              R[illegible]ON = 000000   SV$DUM= 000001          S.STA   000000          S3.PPT= 020000
P.TIMR  000000              RUN     021232R         SV$LOA= 000000          S.STE   000002          S3.PTH= 004000
P.TRIB  000014              RX.NXM= 000020          SYSFDB= ****** GX       S.STEC  000043          S3.RAL= 000010
P.VECT  177732              RX.OVR= 000010          S$$WRG= 000000          S.STLG  000104          S3.RES= 010000
P.VEC1  177752              R$$DER= 000000          S$$YSZ= 007600          S.STLN  000035          S3.RUB= 040000
P.XAQC  000026              R$$K11= 000001          S.ALF   000044          S.STPN  000034          S3.TAB= 000002
P.XAWQ  000030              R$$SND= 000000          S.BLKC  000033          S.TIMC  000054          S3.TME= 002000
P.XBFC  000027              R$$11M= 000000          S.COST= ****** GX       S.TLZ   000012          S3.TSY= 000040
QF$ACT= 000001              R.ADD   000010          S.COUB  000044          S.TMO   000040          S3.8BC= 000100
QF$ALL= 000002              R.CSR   000056          S.COUL= 000040          S.TNRP  000032          S4.ABD= 100000
QF$KNO= 000004              R.FLAG  000012          S.CTL   000006          S.TTEC  000041          S4.ANI= 000400
QF$LOO= 000010              R.FLG   000064          S.DLCF  000102          S.WFA   000016          S4.AVO= 001000
QINS    005636R             R.LEN   000014          S.ERR   000010          S.XDE   000036          S4.BLK= 002000
QUICK   005576R             R.LNK   000000          S.FLG = ****** GX       S.XID   000002          S4.DEC= 004000
Q$$OPT= 000010              R.LST   000104          S.FNCP  000004          S.XMB   000030          S4.DLO= 000100
REMALI  011074R             R.MAP   000110          S.HTNA  000023          S.XMP   000024          S4.EDT= 010000
REMINS  005172R             R.NAM   000002          S.LBE   000044          S.XMT   000012          S4.HFF= 000020
REMNOD  004310R             R.QUE   000062          S.LEN = ****** GX       S.XMTB  000076          S4.HFL= 000007
REMOTE  000132R         002 R.SLN   000070          S.LGR   000031          S.XMTC  000062          S4.HHT= 000040
RE$ADC= 000004              R.SRV   000060          S.LGTH= 000050          S1.DEC= 001000          S4.HSY= 000200
RE$ADF= 000017              R.STA   000066          S.LIN   000004          S1.DPR= 000400          S4.RGS= 020000
RE$ADR= 000007              R.STBL  000210          S.LINK  177776          S1.DSI= 00[illegible]       S4.SFC= 040000
RE$BLK= 000010              SAVCOM  017410R         S.LMA   000026          S1.ESC= 0000[illegible]     S4.VFL= 000010
RE$CAF= 000014              SAVFIL  017376R         S.LMOS  000030          S1.IP[illegible] [illegible]02[illegible]00   S5.ABP= 100000
RE$DAT= 000001              SC$OFF= 000001          S.LMRT  000027          S1.[illegible]BY= 000200     S5.BCC= 020000
RE$DRP= 000016              SC$ON = 000000          S.LMS   000025          S1.OB[illegible] [illegible]00100  S5.DAO= 040000
RE$LDT= 000013              SC$RST= 000003          S.LNK   000000          S1.PTH= [illegible]0010     S5.HPC= 000014
RE$LSN= 000012              SC$SHU= 000002          S.LSA   000005          S1.RES= 010000          S5.HPO= 000020
RE$NML= 000001              SER   = 000001          S.NCUB  000000          S1.RNE= 000020          S5.ITI= 000100
RE$OPE= 000004              SERBIT  001076R         S.NCUL= 000036          S1.RNF= 020000          S5.OXF= 000040
```

N 3

```
VENA       CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 11
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

OF$ON   = 000000           #5-73
OF$SMC  = 000200           #5-73        41-2235
OF.'LU  = 000100            41-2240     41-2243
OF.SMC  = 000200            41-2234     41-2237
OPT1      003656 R          25-1217    #25-1219
OP$ADD  = 000200           #5-73
OP$ALL  = 000200           #5-73
OP$CCS  = 000004           #5-73
OP$CIE  = 004000           #5-73
OP$CON  = 000400           #5-73        47-2701     55-3093     58-3255     73-3966
OP$COP  = 000001           #5-73        41-2231
OP$COS  = 000100           #5-73
OP$DST  = 000200           #5-73
OP$DTE  = 000040           #5-73
OP$DUP  = 000002           #5-73
OP$EST  = 000100           #5-73        25-1189
OP$EVE  = 010000           #5-73        7-181       7-195
OP$FIL  = 000200           #5-73        55-3096     73-3963
OP$GRP  = 000100           #5-73
OP$HOS  = 000001           #5-73        25-1189     59-3333     73-3978
OP$HTM  = 000002           #5-73
OP$INA  = 004000           #5-73        25-1187
OP$INC  = 001000           #5-73        25-1187
OP$INI  = 000000           #5-79
OP$KDS  = 002000           #5-73
OP$KDT  = 000400           #5-73
OP$KEV  = 001000           #5-73        44-2496     45-2558     56-3134     56-3148
OP$KGR  = 001000           #5-73
OP$KNT  = 010000           #5-73
OP$KSK  = 002000           #5-73
OP$LCT  = 000200           #5-73
OP$LNE  = 000040           #5-73        25-1181     25-1191     44-2498     57-3187     57-3192     59-3308     59-3333     71-3897
                            73-3969
OP$LOC  = 000001           #5-73       #5-73
OP$LST  = 000002           #5-73        7-194
OP$LTM  = 000001           #5-73
OP$LVL  = 000010           #5-73        41-2244
OP$MAC  = 000010           #5-73
OP$MCO  = 000002           #5-73
OP$MDE  = 000400           #5-73
OP$MLI  = 000004           #5-73
OP$MON  = 000100           #5-73        47-2696     58-3266
OP$MST  = 000020           #5-73
OP$NAM  = 000004           #5-73       #5-73       #5-73        7-182       7-196       25-1181     25-1193     41-2249
OP$NET  = 004000           #5-73
OP$NLI  = 000002           #5-73
OP$NOD  = 000020           #5-73        7-184       57-3187     59-3308     59-3333     71-3892     73-3972
OP$OUT  = 002000           #5-73        25-1187
OP$OWN  = 000020           #5-73
OP$PAR  = 000010           #5-73
OP$PRI  = 000010           #5-73
OP$RET  = 010000           #5-73        25-1187
```

```
VENA      CREATED BY  MACRO  ON 11-SEP-85 AT 12:08      PAGE 24
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME        REFERENCES

.LTAB            #5-82
.MPLHD           #5-82        5-82
.MXPTB           #5-82
.PECHA           #5-82
.POOL            #5-82
.PRI             #5-82        5-82        5-82
.SCOM            #5-82
.SECSR           #5-82
.SLNB            #5-82         -82
.SLNW            #5-82        5-82
.STNB            #5-82        5-82
.STNW            #5-82
.TIME            #5-82
.UNB             #5-82        5-82
.UNW             #5-82
.VFY             #5-82
.X2CHB           #5-82
.X2CHW           #5-82
.X3CHB           #5-82
.X3CHW           #5-82
```

```
                                      ;**
                                      ; EXPANSION ROUTINES
                                      ;**

                                      ;
                                      ; LOCAL PHYSICAL STATION ADDRESS (BYTE)
                                      ;
000062  005767  000450'               ADDRB:  TST     FLAGS           ; IS IT EITHER MUX OR MTP EXPANSION?
000066  001403                                BEQ     10$             ; BR IF NO
000070  116724  000000G                       MOVB    $$LSAD,(R4)+    ; STORE LOCAL STATION ADDRESS
000074                                        RETURN
000076  112724  177777                10$:    MOVB    #-1,(R4)+       ; DLC LSAD NOT IN MXTAB OR MPTAB FILE
                                      ;
                                      ; LOCAL PHSYICAL STATION ADDRESS (WORD)
                                      ;
000102  005767  000450'               ADDRW:  TST     FLAGS           ; IS IT EITHER MUX OR MTP EXPANSION?
000106  001405                                BEQ     10$             ; BR IF NO
000110  005001                                CLR     R1              ; GET LOCAL STATION ADDRESS
000112  056701  000000G                       BIS     $$LSAD,R1       ; ...
000116  010124                                MOV     R1,(R4)+        ; STORE IT
000120                                        RETURN
000122  012724  177777                10$:    MOV     #-1,(R4)+       ; LSAD NOT IN MXTAB OR MPTAB FILE
000126                                        RETURN
                                      ;
                                      ; ZERO BYTE(S)
                                      ;
000130  112500                        BLKB:   MOVB    (R5)+,R0        ; GET BYTE COUNT
000132  105024                        10$:    CLRB    (R4)+           ; STORE A ZERO BYTE
000134  105300                                DECB    R0              ; LOOP BETWEEN 1 AND 256. TIMES
000136  001375                                BNE     10$
000140                                        RETURN

                                      ;
                                      ; ZERO WORD(S)
                                      ;
000142  112500                        BLKW:   MOVB    (R5)+,R0        ; GET WORD COUNT
000144  005024                        10$:    CLR     (R4)+           ; STORE A ZERO WORD
000146  105300                                DECB    R0              ; LOOP BETWEEN 1 AND 256. TIMES
000150  001375                                BNE     10$
000152                                        RETURN

                                      ;
                                      ; SINGLE BYTE
                                      ;
000154  112524                        BYTE:   MOVB    (R5)+,(R4)+     ; COPY A BYTE
000156                                        RETURN

                                      ;
                                      ; CSR
                                      ;
000160  112500                        CSR:    MOVB    (R5)+,R0        ; GET MODIFICATION VALUE
000162  066700  000000G                       ADD     $$CSR,R0        ; ADD IN CSR ADDRESS
000166  010024                                MOV     R0,(R4)+        ; STORE THE RESULT
000170                                        RETURN

                                      ;
                                      ; COUNTER TIMER
```

```
767 002210                                      RETURN
768
769                                     ;
770                                     ; ACTIVE POLLING RATIO (APR)
771                                     ;
772 002212  016700  000446'             APR:    MOV     INDEX,R0        ; GET CURRENT STATION INDEX
773 002216  116024  000000G                     MOVB    $$APR(R0),(R4)+ ; STORE APR
774 002222                                      RETURN
775
776                                     ;
777                                     ; DEAD POLLING RATIO
778                                     ;
779 002224  116724  000000G             DPRB:   MOVB    $$DPR,(R4)+     ; STORE DPR
780 002230                                      RETURN
781
782 002232  016724  000000G             DPRW:   MOV     $$DPR,(R4)+     ; STORE DPR
783 002236                                      RETURN
```

```
                                .SBTTL  MACRO DEFINITIONS
                        ;***
                        ; LOCAL MACROS
                        ;***


                        ;
                        ; DEFINE TRANSFER TABLE ENTRY
                        ;
                                .MACRO  TTAB$   AA,BB
                                .WORD   BB
                                .ENDM   TTAB$

                        ;***
                        ; LOCAL DATA
                        ;***

 000000                         .PSECT  DATA,D

                        ;
                        ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                        ;
 000000                 TABLE:  TTAB$   ADDRB,NULL      ; EXPANSION ROUTINE IS FIRST ENTRY
 000002                         TTAB$   ADDRW,NULL      ; CLEANUP ROUTINE IS SECOND ENTRY
 000004                         TTAB$   BLKB,POP1
 000006                         TTAB$   BLKW,POP1
 000010                         TTAB$   BYTE,POP1
 000012                         TTAB$   CORE,XCORE
 000014                         TTAB$   CSR,POP1
 000016                         TTAB$   CTIM,NULL
 000020                         TTAB$   DVCHA,NULL
 000022                         TTAB$   EOF,NULL
 000024                         TTAB$   FILE,POP1
 000026                         TTAB$   INT,POP2
 000030                         TTAB$   LIBR,XLIBR
 000032                         TTAB$   LINKS,NULL
 000034                         TTAB$   LFILE,POP3
 000036                         TTAB$   LSTHD,NULL
 000040                         TTAB$   LTAB,NULL
 000042                         TTAB$   MPTAB,POP3
 000044                         TTAB$   MXTAB,POP1
 000046                         TTAB$   NOD,NULL
 000050                         TTAB$   PRI,NULL
 000052                         TTAB$   PRTOV,POP1
 000054                         TTAB$   SCOM,XSCOM
 000056                         TTAB$   SECSR,NULL
 000060                         TTAB$   SLNB,NULL
 000062                         TTAB$   SLNW,NULL
 000064                         TTAB$   STNB,NULL
 000066                         TTAB$   STNW,NULL
 000070                         TTAB$   UNB,NULL
 000072                         TTAB$   UNW,NULL
 000074                         TTAB$   WORD,POP2
 000076                         TTAB$   APR,POP1
 000100                         TTAB$   DPRB,POP1
 000102                         TTAB$   DPRW,POP2
 000104                         TTAB$   TIME,POP2
 000106                         TTAB$   MXPTB,XMXPTB
```

```
                                .SBTTL  MACRO DEFINITIONS
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  GETAD,GETRV,SWTCHI,PUTRC,PUTAD,SAVRG,RESRG,SLTDF$

                        ;***
                        ; LOCAL MACROS
                        ;***
000000                          SLTDF$                  ; DEFINE SLT OFFSETS

                        ;
                        ; DEFINE TRANSFER TABLE ENTRY
                        ;
                                .MACRO  TTAB$   AA,BB
                                .WORD   AA
                                .ENDM   TTAB$

                        ;***
                        ; LOCAL DATA
                        ;***

000000                          .PSECT  DATA,D

                        ;
                        ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                        ;
000000                  TABLE:  TTAB$   ADDRB,NULL      ; EXPANSION ROUTINE IS FIRST ENTRY
000002                          TTAB$   ADDRW,NULL      ; CLEANUP ROUTINE IS SECOND ENTRY
000004                          TTAB$   BLKB,POP1
000006                          TTAB$   BLKW,POP1
000010                          TTAB$   BYTE,POP1
000012                          TTAB$   CORE,XCORE
000014                          TTAB$   CSR,POP1
000016                          TTAB$   CTIM,NULL
000020                          TTAB$   DVCHA,NULL
000022                          TTAB$   EOF,NULL
000024                          TTAB$   FILE,POP1
000026                          TTAB$   INT,POP2
000030                          TTAB$   LIBR,XLIBR
000032                          TTAB$   LINKS,NULL
000034                          TTAB$   LFILE,POP3
000036                          TTAB$   LSTHD,NULL
000040                          TTAB$   LTAB,NULL
000042                          TTAB$   MPTAB,POP3
000044                          TTAB$   MXTAB,POP1
000046                          TTAB$   NOD,NULL
000050                          TTAB$   PRI,NULL
000052                          TTAB$   PRTOV,POP1
000054                          TTAB$   SCOM,XSCOM
000056                          TTAB$   SECSR,NULL
000060                          TTAB$   SLNB,NULL
000062                          TTAB$   SLNW,NULL
000064                          TTAB$   STNB,NULL
000066                          TTAB$   STNW,NULL
000070                          TTAB$   UNB,NULL
000072                          TTAB$   UNW,NULL
```

```
   676                                          .SBTTL  X3CH
   677
   678                                  ;
   679                                  ; DEFAULT BLOCK SIZE
   680                                  ;
   681 001674  005723                   X3CH01: TST     (R3)+
   682 001676                                   RETURN
   683                                  ;
   684                                  ; MAXIMUM BLOCK SIZE
   685                                  ;
   686 001700  005723                   X3CH02: TST     (R3)+
   687 001702                                   RETURN
   688                                  ;
   689                                  ; DEFAULT WINDOW SIZE
   690                                  ;
   691 001704  005203                   X3CH03: INC     R3
   692 001706                                   RETURN
   693                                  ;
   694                                  ; MAXIMUM WINDOW SIZE
   695                                  ;
   696 001710  005203                   X3CH04: INC     R3
   697 001712                                   RETURN
   698                                  ;
   699                                  ; CALL TIMER
   700                                  ;
   701 001714  005203                   X3CH05: INC     R3
   702 001716                                   RETURN
   703                                  ;
   704                                  ; CLEAR TIMER
   705                                  ;
   706 001720  005203                   X3CH06: INC     R3
   707 001722                                   RETURN
   708                                  ;
   709                                  ; RESET TIMER
   710                                  ;
   711 001724  005203                   X3CH07: INC     R3
   712 001726                                   RETURN
   713                                  ;
   714                                  ; RESTART TIMER
   715                                  ;
   716 001730  005203                   X3CH08: INC     R3
   717 001732                                   RETURN
   718                                  ;
   719                                  ; MAXIMUM CLEAR VALUE
   720                                  ;
   721 001734  005203                   X3CH09: INC     R3
   722 001736                                   RETURN
   723                                  ;
   724                                  ; MAXIMUM RESETS VALUE
   725                                  ;
   726 001740  005203                   X3CH10: INC     R3
   727 001742                                   RETURN
   728                                  ;
   729                                  ; MAXIMUM RESTARTS VALUE
   730                                  ;
   731 001744  005203                   X3CH11: INC     R3
   732 001746                                   RETURN
```

```
    126                                          .SBTTL  ERROR MESSAGES
    127
    128 000046                                   NTLER$  ,00,8B,CERR,RTSPC,CFLIN,<network name parameter>           ;**-2
    129 000106                                   NTLER$  ,03,8B,CERR,RTSPC,CFLIN,<DTE address value>
    130 000140                                   NTLER$  ,04,8B,CERR,RTSPC,CFLIN,<hash table size>
    131 000170                                   NTLER$  ,05,8B,CERR,RTSPC,CFLIN,<counter timer value>
    132 000224                                   NTLER$  ,06,8B,CERR,RTSPC,CFLIN,<line-id>
    133
    134                                          .LIST   BEX                                                        ;**-1
    135                                          .EVEN
    136 000000                                   .PSECT
```

```
                                                                 M 12
VFGDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:28  Page 20-1
Symbol table

A$$CHK= 000000        HF$XDF= 000020        LINID1  000146R   003 L$NRFS  000100        P$$P45= 000000
A$$CPS= 000000        HSHMN = 000040        LINNAM  000042R   002 L$NRPK  000056        P$$WRD= 000000
A$$PRI= 000000        HSHMX = 001000        LINUNT  000045R   002 L$NRRE  000105        Q$$OPT= 000010
A$$TRP= 000000        HSHSZ = ****** GX     LN$OFF= 000001        L$NRST  000107        RTSPC = ****** GX
BIASX   000576R       HSHTBL  000262R       LN$ON = 000000        L$NTBY  000052        R$R0  = 000002
CERR  = ****** GX     H$CUG   000010        LN$SHU= 000002        L$NTCA  000070        R$R1  = 000004
CFGBF = ****** GX     H$DST   000012        LN.CLO= 000000        L$NTFS  000102        R$R2  = 000006
CFGSZ = ****** GX     H$D29   000014        LN.DUM= 000005        L$NTPK  000062        R$R3  = 000010
CFLIN = ****** GX     H$FLG   000000        LN.LOA= 000004        L$OMST  000002        R$R4  = 000012
CHKDTE  000426R       H$GLEN  000104        LN.LOO= 000003        L$PCS   000112        R$R5  = 000014
CNTTIM  000332R       H$GLT   000044        LN.OAU= 000003        L$PLS   000110        R$$DER= 000000
COUNT   000035R   002 H$GNAM  000050        LN.OFF= 000001        L$QCNT  000122        R$$K11= 000001
CVTBUF  000036R   002 H$GNML= 000020        LN.ON = 000000        L$QUEH  000124        R$$SND= 000000
C$$CKP= 000000        H$GPT   000046        LN.OOP= 000004        L$QUET  000130        R$$11M= 000000
C$$ORE= 000400        H$HITS  000034        LN.OPE= 000001        L$RTRY  000007        SF.ACT= 000200
C$$RSH= 177564        H$HLEN  000044        LN.REF= 000002        L$SLN   000004        SF.ENA= 000100
DEVCTL  000644R       H$LBDA  000070        LN.SER= 000002        L$ST    000005        SF.LPB= 000004
DEVNAM  000616R       H$LBDN  000072        LN.STA= 000017        L$TCLZ  000044        SF.MFL= 000040
DEVUNT  000662R       H$LDTE  000002        LN.SUB= 000360        L$TIM   000006        SF.PAC= 000020
DE.OFF= ****** GX     H$LEN   000042        LN.TRI= 000006        L$$ASG= 000000        SF.REA= 000010
DE.ON = ****** GX     H$LOTS  000032        LP$EIP= 002000        L$$DRV= 000000        SF.SER= 000001
DTEAD   000012R   003 H$NETW  000024        LP$ENB= 004000        L$$P11= 000001        SF.SVC= 000002
DTEADD  000000R   002 H$NML = 000006        LP$PCT= 001400        L$$11R= 000000        SF.UNL= 000040
DTEAEN  000140R       H$NPT   000022        LP$TMR= 100000        L.COST  000015        SPACE = 000040
DTEALN= 000017        H$PTB   000020        LP$UP = 010000        L.CTL   000012        SPFILL= ****** GX
DTEAST  000052R       H$PVC   000006        LP$WTD= 020000        L.CVA   177776        STRNXT  000706R
DTEDES= ****** GX     H$RDTE  000004        LP$WTS= 040000        L.DDM   000002        SYNERR= ****** GX
DTEDF   000000R   003 H$RNW   000042        LS$GDN= 000004        L.DDS   000004        SYSFDB= ****** GX
DTEDIG  000114R       H$SVC   000036        LS$OFF= 000000        L.DLC   000003        S$$WRG= 000000
DTEEND  000410R       H$TRB   000016        LS$STR= 000002        L.DLM   000006        S$$YSZ= 007600
DTEFLG= ****** GX     H$XAVL  000100        LS$UP = 000003        L.DLS   000010        S.COST  000001
DTEKW   000000RG  004 H$XBIA  000074        LS$WT = 000001        L.FLG   000000        S.FLG   000000
DTEND = 000016R   002 H$X29C  000040        L$ACHN  000012        L.KRBA  000016        S.LEN   000004
DTENET  000350R       I$$RAR= 000000        L$APVC  000014        L.LEN = 000022        S.NMST  000002
DTENW   000076R   003 I$$RDN= 000000        L$ASVC  000010        L.MPF   000022        S.OWNR  000003
DTEPCK  000017R   002 K$$CNT= 177546        L$AUC   000042        L.NMST  000020        T$$KMG= 000000
DTEST   000000RG  003 K$$CSR= 177546        L$CHLS  000034        L.NSTA  000014        T$$MIN= 000000
DTNMMX= 000006        K$$LDC= 000000        L$CHTB  000030        L.OWNR  000021        V$$CTR= 001000
DTNTNM  000027R   002 K$$TPS= 000074        L$CLEN= 000044        L.UNT   000013        X$$DBT= 000000
D$$PUG= 177514        LD$LP = 000000        L$CTEN  000032        M$$CRB= 000124        $ALPHA= 000022
D$$ISK= 000000        LF.ACT= 100000        L$CTIM  000040        M$$CRX= 000000        $ANY  = 000020
D$$L11= 000001        LF.BRO= 000400        L$DTEA  000020        M$$FCS= 000000        $BFR  = ****** GX
D$$YNC= 000000        LF.BWT= 000007        L$DTEL  000017        M$$MGE= 000000        $BLANK= 000006
D$$YNM= 000000        LF.ENA= 002000        L$GLEN  000134        M$$NET= 000000        $CAT5 = ****** GX
END     000114R   003 LF.LPB= 001000        L$LEN   000122        M$$OVR= 000000        $CDTB = ****** GX
E$$XPR= 000000        LF.MDC= 000100        L$LLCH  000016        NEXT    000040R   002 $CEACC= ****** GX
FMT8B = ****** GX     LF.MFL= 004000        L$LNK   000000        NUMDTE= 000020        $DIGIT= 000024
FM.8B = 000000        LF.MTP= 000020        L$MCHN  000036        N$$ACC= 000001        $DNUMB= 000014
F$$LVL= 000001        LF.PAC= 000200        L$NETW  000114        N$$BUF= 000001        $EOS  = 000012
F.RSIZ= ****** GX     LF.RDY= 040000        L$NIRE  000072        N$$LDV= 000001        $ERR00  000046R   002
G$$TPP= 000000        LF.REA= 010000        L$NLRE  000104        N$$MCP= 000001        $ERR03  000106R   002
G$$TSS= 000000        LF.SER= 000040        L$NMAC  000076        N$$MLL= 000001        $ERR04  000140R   002
G$$TTK= 000000        LF.TIM= 000010        L$NMAS  000074        N$$MOV= 000010        $ERR05  000170R   002
G$$WRD= 000000        LF.UNL= 020000        L$NMST  000003        N$$NCT= 000001        $ERR06  000224R   002
HF$DLM= 000002        LF.X2P= 000000        L$NNRE  000106        N$$PFM= 000001        $ERR1T= ****** GX
HF$GWY= 000010        LINCTL  000044R   002 L$NRBY  000046        PCKBCD  000756R       $EXIT = 000000
HF$HOS= 000004        LINID   000124R   003 L$NRCA  000066        PF.XDF= ****** GX     $FAIL = 177777
```

```
    161                                            .SBTTL  ERROR MESSAGES
    162
    163                                            .ENABL  LC
    164
    165 000064                                     NTLER$  ,N0,8B,CERR,RTSPC,CFLIN,<circuit identification>
    166 000124                                     NTLER$  ,N1,8B,CERR,RTSPC,CFLIN,<process owner name>
    167 000160                                     NTLER$  ,N2,8B,CERR,RTSPC,CFLIN,<flags byte value>
    168 000212                                     NTLER$  ,N3,8,CERR,RTSPC,CFLIN,<Resource allocation failure>
    169 000256                                     NTLER$  ,P8,8B,CERR,RTSPC,CFLIN,<maximum block size>
    170 000312                                     NTLER$  ,00,8B,CERR,RTSPC,CFLIN,<maximum window size>
    171 000346                                     NTLER$  ,05,8B,CERR,RTSPC,CFLIN,<counter timer value>
    172 000402                                     NTLER$  ,09,8B,CERR,RTSPC,CFLIN,<logical channel number>
    173                                            .DSABL  LC
    174                                            .EVEN
    175                                            .LIST   BEX
    176 000000                                     .PSECT
    687                                                 .SBTTL  DLMTLC - SET UP TRANSPORT LINE COUNTERS
    688
    689                                         ;+
    690                                         ;
    691                                         ; DLMTLC - SET UP TRANSPORT LINE COUNTERS
    692                                         ;
    693                                         ; INPUTS:
    694                                         ;       R1 - PORT NUMBER
    695                                         ;
    696                                         ; OUTPUTS:
    697                                         ;       PORT NUMBER STORED IN TRANSPORT LINE COUNTER BLOCK
    698                                         ;
    699                                         ;       R0,R1,R2,R3,R4 DESTROYED
    700                                         ;
    701                                         ;       NOTE: THIS ROUTINE MUST BE CALLED FROM SYSTEM STATE
    702                                         ;-
    703 001662  016703  000000G                 DLMTLC: MOV     $SLTMA,R3               ; GET ADDRESS OF VECTOR TABLE
    704 001666  022367  000000G                 5$:     CMP     (R3)+,$SLTA             ; IS THIS THE CORRECT ENTRY?
    705 001672  001375                                  BNE     5$                      ; BR IF NO
    706 001674  005743                                  TST     -(R3)                   ; BACK UP TO CORRECT ENTRY
    707 001676  166703  000000G                         SUB     $SLTMA,R3               ; GET SYSTEM LINE NUMBER*2
    708 001702  010367  000030'                         MOV     R3,DLMSLN               ; SAVE SYSTEM LINE NUMBER FOR LATER
    709 001706  006267  000030'                         ASR     DLMSLN                  ; ...
    710 001712  066703  000000G                         ADD     $LLCTA,R3               ; INDEX INTO REVERSE MAPPING TABLE
    711 001716  011303                                  MOV     (R3),R3                 ; GET CHANNEL NUMBER AND PDV INDEX
    712 001720  100004                                  BPL     10$                     ; BR IF NOT MULTIPOINT
    713 001722  066703  000026'                         ADD     DLMSTA,R3               ; POINT TO CORRECT STATION
    714 001726  006303                                  ASL     R3                      ; GET ADDRESS OF STATION TABLE
    715 001730  011303                                  MOV     (R3),R3                 ; GET CHANNEL NUMBER AND PDV INDEX
    716 001732  005046                          10$:    CLR     -(SP)                   ; GET INDEX INTO FORWARD MAPPING TABLE
    717 001734  110316                                  MOVB    R3,(SP)                 ; ...
    718 001736  006316                                  ASL     (SP)                    ; ...
    719 001740  012603                                  MOV     (SP)+,R3                ; RETRIEVE FORWARD MAPPING TABLE INDEX
    720 001742  016704  000000G                         MOV     $PDVTA,R4               ; GET ADDRESS OF PDV VECTOR TABLE
    721 001746  016700  000000G                         MOV     $PDVNM,R0               ; GET NUMBER OF PDV'S IN SYSTEM
    722 001752  012402                          15$:    MOV     (R4)+,R2                ; GET NEXT PDV IN LIST
    723 001754  022762  114224  000000G         CMP     #^RXPT,Z.NAM(R2)        ; IS THIS XPT?
    724 001762  001403                                  BEQ     17$                     ; BR IF YES
    725 001764                                          SOB     R0,15$                  ; CONTINUE UNTIL ALL CHECKED
    726 001770  000470                                  BR      60$                     ; XPT NOT IN SYSTEM?
    727 001772  060203                          17$:    ADD     R2,R3                   ; POINT TO FORWARD MAPPING TABLE ENTRY
    728 001774  062703  000000G                         ADD     #Z.MAP,R3               ; ...
    729 002000  011304                                  MOV     (R3),R4                 ; GET SLN/STA PAIR
    730 002002                                          SAVMAP                          ; SAVE CURRENT MAPPING
    731 002006                                          GETRV   Z.DAT(R2),#N$XLEN       ; READ IN XPT'S DDB
    732 002026                                          MAP     N$TLC+2+$BFR            ; MAP TO CIRCUIT COUNTERS
    733 002034  016703  000100G                         MOV     N$TLC+$BFR,R3           ; GET NUMBER OF LINE COUNTERS
    734 002040  016702  000104G                         MOV     N$TLC+4+$BFR,R2         ; GET ADDRESS OF FIRST COUNTER BLOCK
    735
    736 002044                                  20$:    GETAD   R2,#T$LLEN              ; READ IN COUNTER BLOCK
    737 002062  005767  000000G                         TST     T$LIN+$BFR              ; HAS THIS BLOCK BEEN ASSIGNED ALREADY?
    738 002066  100410                                  BMI     30$                     ; BR IF NO - USE IT
    739 002070  026704  000000G                         CMP     T$LIN+$BFR,R4           ; ELSE IS THIS THE ONE WE WANT?
    740 002074  001407                                  BEQ     40$                     ; BR IF YES
    741 002076  062702  000046                          ADD     #T$LLEN,R2              ; POINT TO NEXT COUNTER BLOCK
    742 002102                                          SOB     R3,20$                  ; CHECK ALL COUNTER BLOCKS
    743 002106  000417                                  BR      50$                     ; BR IF ONE NOT AVAILABLE
```

Table of contents

```
     5-   54    MACRO DEFINITIONS
     6-  111    LOCAL DATA
     7-  137    ERROR MESSAGES
     8-  154    $QSVC - LOOK FOR SVC$DF MACRO
     9-  183    SVC$DF STATE TABLE
    10-  257    SVC$DF ACTION ROUTINES
    14-  493    FNDOWN - FIND SVC OWNER'S PDV INDEX
    15-  527    SETTLC - SET UP TRANSPORT LINE COUNTERS
    16-  596    PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
    17-  633    STRNXT - STORE NEXT DIGIT
```

```
                                          .SBTTL  FNDOWN - FIND SVC OWNER'S PDV INDEX

                                  ;+
                                  ;
                                  ; FNDOWN - FIND PDV INDEX OF SVC OWNER
                                  ;
                                  ; INPUTS:
                                  ;       R1 - RAD50 PROCESS NAME OF SVC OWNER
                                  ;
                                  ; OUTPUTS:
                                  ;       CARRY CLEAR:
                                  ;          R2 - PDV INDEX OF PROCESS
                                  ;       CARRY SET:
                                  ;          PDV INDEX OF OWNER NOT FOUND
                                  ;
                                  ;-
001214  010146                    FNDOWN: MOV     R1,-(SP)                ; SAVE NAME TO BE MATCHED
001216  016701  000000G                   MOV     $PDVTA,R1               ; GET PDV TABLE ADDRESS
001222  016702  000000G                   MOV     $PDVNM,R2               ; AND NUMBER OF PDV'S
001226  001407                            BEQ     30$                     ; IF ZERO, NO PDV'S!
001230  012100                    10$:    MOV     (R1)+,R0                ; GET PDV ADDRESS
001232  001403                            BEQ     20$                     ; IF ZERO, SKIP THIS ONE
001234  026016  000000G                   CMP     Z.NAM(R0),(SP)          ; DOES THE NAME MATCH?
001240  001403                            BEQ     40$                     ; IF EQ, YES (C-BIT CLEAR)
001242                            20$:    SOB     R2,10$                  ; LOOP $PDVNM TIMES
001246  000261                    30$:    SEC                             ; INDICATE FAILURE
001250  005226                    40$:    INC     (SP)+                   ; PURGE STACK
001252  006016                            ROR     (SP)                    ; SAVE C-BIT
001254  005741                            TST     -(R1)                   ; CALCULATE PDV INDEX
001256  166701  000000G                   SUB     $PDVTA,R1
001262  006316                            ASL     (SP)                    ; RESTORE C-BIT
001264  110167  000020'                   MOVB    R1,OWNSVC               ; SAVE OWNER PDV INDEX
001270                                    RETURN                          ;
```

```
                                   .SBTTL  CHKEVT - CHECK THE SPECIFIED EVENT MASKS

                            ;+
                            ;
                            ; CHKEVT - CHECK THE EVENT FILTER MASKS TO SEE IF ANY MASKS ARE
                            ;          STILL SET
                            ;
                            ; INPUTS:
                            ;       R5 - ADDRESS OF END OF EVENT FILTER MASKS TO OPERATE ON
                            ;
                            ; OUTPUTS:
                            ;       CARRY CLEAR:
                            ;          EVENT MASKS ARE STILL SET IN THE FILTER BLOCK
                            ;       CARRY SET:
                            ;          THE FILTER BLOCK DOES NOT CONTAIN ANY MASKS THAT ARE SET
                            ;       R3,R5 DESTROYED
                            ;
                            ;-

017352  005745              CHKEVT: TST     -(R5)                   ; ARE ANY EVENT MASKS SET?
017354  001007                      BNE     10$                     ; BR IF YES
017356  005745                      TST     -(R5)                   ; ...
017360  001005                      BNE     10$                     ; ...
017362  005745                      TST     -(R5)                   ; ...
017364  001003                      BNE     10$                     ; ...
017366  005745                      TST     -(R5)                   ; ...
017370  001001                      BNE     10$                     ; ...
017372  000261                      SEC                             ; ELSE INDICATE NO MASKS SET
017374                      10$:    RETURN
```

```
                                                   .SBTTL  $VHST - SET/CLEAR HOST ADDRESS
                                         ;+
                                         ; **-$VHST- SET/CLEAR HOST
                                         ;
                                         ; OUTPUTS:
                                         ;          HOST ADDRESS UPDATED IN CEXCM
                                         ;-

020662                                   $VHST::
020662                                             GETRV   $DECPT,#D$END             ; READ DECNET HOME BLOCK

020702  032767  000000G 000000G                    BIT     #CM$SET,$CMAND            ; IS THIS A SET OPERATION ?
020710  001404                                     BEQ     10$                       ; IF EQ, NO .. MUST BE CLEAR
020712  016767  000040G 000022G                    MOV     RQB+LO.HOS,$BFR+D$HOST    ; ELSE, STORE HOST NODE ADDRESS
020720  000403                                     BR      20$                       ; AND LEAVE

020722  016767  000014G 000022G 10$:               MOV     $BFR+D$LNUM,$BFR+D$HOST ; SET LOCAL NODE AS HOST NODE

020730                                   20$:      PUTRC                             ; WRITE BACK DECNET HOME BLOCK

020734                                             RETURN
```

```
                                                                                   N 3
VENA - VNP ENABLE/DISABLE LINES MACRO V05.03b Wednesday 11-Sep-85 12:06  Page 85-5
Symbol table

S5.RPO= 002000          T$LOPR  000002          T.RDAT  000034          UC.NPR= 000100          U.CTYP  000052
S5.SW1= 000001          T$LTCL  000024          T.RRFL  000072          UC.PWF= 000020          U.CW1   000010
S5.TMM= 000002          T$LTIM  000026          T.RTM   000055          UC.QUE= 000040          U.CW2   000012
S5.VER= 010000          T$LTPR  000014          T.RTY   000030          UD.UNS= 000000          U.CW3   000014
S5.XOF= 000004          T$LTPS  000020          T.SAST  000054          UD.160= 000004          U.CW4   000016
S5.XON= 000010          T$NAPL  000004          T.SEL   000040          UD.200= 000001          U.CYL = 000104
S6.EIO= 000400          T$NFE   000000          T.SELT  000052          UD.556= 000002          U.DCB   000000
S6.RDI= 100000          T$NLEN  000010          T.SFNC  000014          UD.625= 000005          U.FCDE= 000042
S6.RLU= 001000          T$NNUL  000002          T.SITC  000072          UD.8K = 000006          U.FNUM= 000040
TF.FFE= 000004          T$NOPL  000006          T.SRCT  000071          UD.800= 000003          U.GRP = 000102
TF.ROF= 000001          T$NRNI  000042          T.STAT  000032          UM.CLI= 000036          U.KCSR= 000032
TF.RTO= 000002          T$NRPL  000005          T.ST2   000034          UM.DSB= 000200          U.KCS6= 000034
TF.TAE= 000010          T$NRUL  000007          T.ST3   000036          UM.NBR= 000400          U.LUIC  177774
TF.TOF= 000004          T$NVR   000001          T.TCBL  000030          UM.OVR= 000001          U.MEDI= 000040
TF.TTO= 000010          T$RPRI  000040          T.TIMC  000016          US.ABO= 000001          U.MLUN= 000054
TF.URE= 000001          T$SVC   000034          T.TIO   000057          US.BSP= 000002          U.MUP   177772
TF.USA= 000002          T$T5    000030          T.TKSZ  000060          US.BSY= 000200          U.OWN   177776
TICKS = 000001          T$T6    000032          T.TMR   000000          US.FOR= 000040          U.RBNS= 000112
TMPMSK  000076R     002 T$$KMG= 000000          T.TOD   000070          US.FRK= 000002          U.RCTC= 000113
TR.ALF  000102          T$$MIN= 000000          T.TRLG= 000076          US.KPF= 000001          U.RCTS= 000110
TR.DEV  000100          T.ACT   000031          T.TRLN  000005          US.LAB= 000004          U.RED   000002
TR.LEN= 000210          T.ACTL  000052          T.TRPN  000004          US.MDE= 000002          U.RED2  000034
TR.MST  000074          T.ASTL  000016          T.UCB   000026          US.MDM= 000020          U.SCB   000020
TR.PRE  000076          T.ATT   000062          T.XASQ  000010          US.MNT= 000100          U.SHST= 000076
TR.TLZ  000072          T.BTTC  000074          T.XBTC  000024          US.OFL= 000001          U.SHUN= 000074
TSKSCH  021630R         T.BUF   000006          T.XDAT  000030          US.PUB= 000004          U.SPC = 000036
TS.BLK= 171700          T.CHA   000032          T2.ABO= 000100          US.PWF= 000010          U.STS   000005
TS.CKP= 000200          T.CPCB  000004          T2.AST= 100000          US.RED= 000002          U.ST2   000007
TS.CKR= 000100          T.CSR   000022          T2.CAF= 000400          US.SHR= 000001          U.SUB = 000036
TS.EXE= 100000          T.DELT  000056          T2.CHK= 020000          US.SPU= 000002          U.TCHP  000042
TS.HLD= 002000          T.DIN   000044          T2.CKD= 010000          US.UMD= 000010          U.TCVP  000043
TS.MSG= 020000          T.DOUT  000042          T2.DST= 040000          US.VV = 000001          U.TFLK  000040
TS.NRP= 010000          T.DPRI  000040          T2.FXD= 002000          US.WCK= 000010          U.TFRQ  000036
TS.OUT= 000400          T.EFLG  000022          T2.HLT= 000200          UU.ABO= 000400          U.TIXL  000060
TS.RDN= 040000          T.ERRC  000020          T2.REX= 001000          UU.ATN= 000100          U.TLPP  000053
TS.RUN= 004000          T.IOC   000003          T2.SEF= 004000          UU.AVN= 000004          U.TMTI  000047
TS.STP= 001000          T.ITM   000001          T2.SPN= 000004          UU.GUS= 000010          U.TRCK= 000100
TX.NXM= 000002          T.LBF   000046          T2.STP= 000020          UU.IOS= 002000          U.TSTA  000026
TX.OVR= 000001          T.LBN   000041          T2.WFR= 000001          UU.ONL= 000020          U.TST5  000054
TX.TMO= 000004          T.LDV   000044          T3.ACP= 100000          UU.RCT= 000002          U.TST6  000056
TY$CHA= 000400          T.LNK   000000          T3.CAL= 000100          UU.RDY= 000200          U.TTAB  000050
TY$EVE= 001000          T.LTM   000054          T3.CLI= 001000          UU.SER= 000001          U.TTYP  000046
TY$STA= 002000          T.MXSZ  000050          T3.GFL= 000010          UU.SIO= 001000          U.TUX   000024
TY$SUM= 000000          T.NAM   000006          T3.MCR= 004000          UU.SPC= 000040          U.UHVR= 000107
T$FLAG  000044          T.NDM   000066          T3.NET= 000020          U.ACB   000062          U.UIC   000044
T$LIF   000013          T.OFF   000066          T3.NSD= 000200          U.ACP = 000032          U.UNFL= 000052
T$LIFL  000013          T.PCB   000046          T3.PMD= 040000          U.ADMA  000066          U.UNIT  000006
T$LIFO  000013          T.PIND  000007          T3.PRV= 010000          U.AFLG  000064          U.UNSZ= 000114
T$LIFS  000013          T.PRI   000002          T3.REM= 020000          U.ATT   000022          U.UNTI= 000060
T$LIN   000000          T.QRAC  000060          T3.ROV= 000040          U.BPKT= 000044          U.USVR= 000106
T$LIPS  000006          T.QRIN  000062          T3.RST= 000400          U.BUF   000024          U.UTIL= 000036
T$LLD   000012          T.QRPD  000064          T3.SLV= 002000          U.CBF = 000032          U.VCB = 000034
T$LLDC  000045          T.QUE   000024          T3.SWS= 000002          U.CLI   177772          U.VSER= 000120
T$LLDL  000012          T.QU2 = 000026          UC.ALG= 000200          U.CNT   000030          U.2MED= 000070
T$LLDO  000012          T.RBF   000050          UC.ATT= 000010          U.CUTQ  000030          U2.AT.= 000020
T$LLDS  000012          T.RBTC  000020          UC.KIL= 000004          U.CTCB  000026          U2.CRT= 002000
T$LLEN  000046          T.RCVL  000012          UC.LGH= 000003          U.CTL   000004          U2.DH1= 100000
```

```
                                                                  N  4
VENA      CREATED BY  MACRO  ON 11-SEP-85 AT 12:08     PAGE 12
SYMBOL CROSS REFERENCE                                 CREF   04.00

SYMBOL   VALUE          REFERENCES

OP$RPA  = 000010        #5-73       25-1189
OP$SCO  = 000001        #5-73       42-2311   42-2361   43-2467
OP$SEG  = 000400        #5-73       25-1187
OP$SER  = 000400        #5-73
OP$SIN  = 000010        #5-73       7-183     59-3308   59-3333   71-3905   73-3975
OP$STA  = 000004        #5-73
OP$TER  = 000001        #5-79
OP$TPA  = 000020        #5-73       25-1189
OP$TRI  = 000040        #5-73
OP$UCS  = 000020        #5-73
OP$USE  = 000002        #5-73       41-2238
OP$VEC  = 000040        #5-73
OP$VER  = 000040        #5-73       25-1189
OP$XAC  = 000001        #5-73
OP$XPR  = 000002        #5-73
OP$X5S  = 000004        #5-73
OP$X9S  = 000010        #5-73
O.FLG     000003        *41-2234  *41-2237  *41-2240  *41-2243  *41-2247  *41-2248  *41-2265
O.LEN     000012          29-1620   41-2221   41-2258   41-2273
O.MXC     000004        *41-2233  *41-2266
O.NAM     000006        *41-2251  *41-2252  *41-2267  *41-2268
O.TYP     000002          29-1621 *41-2264    41-2275
PDVX      000072 R      #6-137    *39-1951  *40-2096
PH$HDE  = 000004        #5-79
PH$LOC  = 000002        #5-79
PH$MTS  = 000003        #5-79
PH$UMP  = 000000        #5-79
PH$WCS  = 000001        #5-79
PLBFLG    000122 R      #6-143
PROEVT    012422 R        45-2568 #46-2638
PRVUNM    000166 R      #6-160    *29-1616    29-1623 *30-1644    30-1652 *60-3375    60-3401 *61-3428    61-3458
PVALID    000240 R      #7-181      44-2507
P$LEN     000052          74-4015   74-4055   75-4106
P.LEN   = 000154          22-1031
P.TRIB    000014          22-1033
QF$ACT  = 000001        #5-73
QF$ALL  = 000002        #5-73       42-2307
QF$KNO  = 000004        #5-73       41-2203   41-2210   41-2223   42-2307   43-2465   47-2689   55-3090   56-3126
                          56-3150   58-3244   73-3960

QF$LOO  = 000010        #5-73
QINS      005636 R        28-1580 #32-1706    41-2278   59-3347
QUICK     005576 R      #31-1675
REMALI    011074 R        42-2316 #42-2319    42-2376   43-2473
REMINS    005172 R        27-1460 #28-1569
REMNOD    004310 R        27-1395 #27-1398    28-1497
REMOTE    000132 R      #6-150    *54-3063  *54-3067    61-3451   71-3894
RE$ADC  = 000004        #5-79
RE$ADF  = 000017        #5-79
RE$ADR  = 000007        #5-79
RE$BLK  = 000010        #5-79
RE$CAF  = 000014        #5-79
RE$DAT  = 000001        #5-79
```

```
                                                                 N 5
**FILE**ID**VEXP




VV      VV   EEEEEEEEEE   XX      XX   PPPPPPPP
VV      VV   EEEEEEEEEE   XX      XX   PPPPPPPP
VV      VV   EE           XX      XX   PP      PP
VV      VV   EE           XX      XX   PP      PP
VV      VV   EE             XX  XX     PP      PP
VV      VV   EE             XX  XX     PP      PP
VV      VV   EEEEEEEE         XX       PPPPPPPP
VV      VV   EEEEEEEE         XX       PPPPPPPP
VV      VV   EE             XX  XX     PP
VV      VV   EE             XX  XX     PP
  VV  VV     EE           XX      XX   PP               ....
  VV  VV     EE           XX      XX   PP               ....
    VV       EEEEEEEEEE   XX      XX   PP               ....
    VV       EEEEEEEEEE   XX      XX   PP               ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                 ;
000172  016724  000000G          CTIM:   MOV     $CTIM,(R4)+         ; STORE THE COUNTER TIMER VALUE
000176                                   RETURN

                                 ;
                                 ; DEVICE CHARACTERISTICS
                                 ;
000200  005000                   DVCHA:  CLR     R0                  ; ASSUME DLC
000202  032767  000000G 000000G          BIT     #FL.DDM,$FLAGS      ; IS IT A DDM?
000210  001414                           BEQ     10$                 ; IF EQ, NO
000212  032767  000000G 000000G          BIT     #FL.MUX,$FLAGS      ; IS IT A MUX DEVICE?
000220  001410                           BEQ     10$                 ; IF EQ, NO
000222  005767  000450'                  TST     FLAGS               ; IS IT MUX EXPANSION?
000226  100023                           BPL     101$                ; IF PL, NO
000230  016700  000446'                  MOV     INDEX,R0            ; GET SECONDARY FILE COUNTER
000234  006300                           ASL     R0                  ; CONVERT TO A DOUBLEWORD INDEX
000236  006300                           ASL     R0                  ;           ...
000240  022020                           CMP     (R0)+,(R0)+         ; SKIP DLC CHARACTERISTICS
000242                           10$:    SAVRG   R1                  ; SAVE REG
000244  016701  000000G                  MOV     $SLTA,R1            ; COPY SLT ADDRESS
000250  010461  177776                   MOV     R4,L.CVA(R1)        ; STORE DEV CHAR POINTER
000254  066761  000000G 177776           ADD     $LINK,L.CVA(R1)     ; ...
000262                                   RESRG   R1                  ; RESTORE REG

000264  016024  000000G                  MOV     $$DCHA(R0),(R4)+ ; COPY DEVICE CHARACTERISTICS
000270  016024  000002G                  MOV     $$DCHA+2(R0),(R4)+ ; TWO WORDS
000274                                   RETURN
000276  012724  177777           101$:   MOV     #-1,(R4)+           ; DDM DVCHA NOT IN MXTAB FILE
000302  012724  177777                   MOV     #-1,(R4)+           ;         ...
000306                                   RETURN

                                 ;
                                 ; PROTOCOL EMULATOR CHARACTERISTICS
                                 ;
000310  005000                   PECHA:  CLR     R0                  ; ASSUME DLC
000312  032767  000000G 000000G          BIT     #FL.DDM,$FLAGS      ; IS IT A DDM?
000320  001413                           BEQ     10$                 ; IF EQ, NO
000322  032767  000000G 000000G          BIT     #FL.MUX,$FLAGS      ; IS IT A MUX DEVICE?
000330  001407                           BEQ     10$                 ; IF EQ, NO
000332  005767  000450'                  TST     FLAGS               ; IS IT MUX EXPANSION?
000336  100004                           BPL     10$                 ; IF PL, NO
000340  016700  000446'                  MOV     INDEX,R0            ; GET SECONDARY FILE COUNTER
000344  006300                           ASL     R0                  ; CONVERT TO A WORD INDEX
000346  005720                           TST     (R0)+               ; SKIP DLC CHARACTERISTICS
000350  016024  000000G          10$:    MOV     $$PCHA(R0),(R4)+ ; COPY DEVICE CHARACTERISTICS
000354                                   RETURN

                                 ;
                                 ; END OF FILE
                                 ;
000356                           EOF:    CRASH

                                 ;
                                 ; LOGICAL LINKS
                                 ;
000360                           LINKS:  SWTCHI  #$HMBUF             ; SET INPUT BUFFER ADDRESS
000366                                   GETRV   $DECPT,#D$END       ; READ DECNET HOME BLOCK
```

```
                                        .SBTTL  X3CH

                                ;
                                ; DEFAULT BLOCK SIZE
                                ;
 002240  016724  000000G        X3CH01: MOV     $X3DBS,(R4)+            ; STORE DEFAULT BLOCK SIZE
 002244                                 RETURN
                                ;
                                ; MAXIMUM BLOCK SIZE
                                ;
 002246  016724  000000G        X3CH02: MOV     $X3MBS,(R4)+            ; STORE MAXIMUM BLOCK SIZE
 002252                                 RETURN
                                ;
                                ; DEFAULT WINDOW SIZE
                                ;
 002254  116724  000000G        X3CH03: MOVB    $X3DWS,(R4)+            ; STORE DEFAULT WINDOW SIZE
 002260                                 RETURN
                                ;
                                ; MAXIMUM WINDOW SIZE
                                ;
 002262  116724  000000G        X3CH04: MOVB    $X3MWS,(R4)+            ; STORE MAXIMUM WINDOW SIZE
 002266                                 RETURN
                                ;
                                ; CALL TIMER
                                ;
 002270  116724  000000G        X3CH05: MOVB    $X3CAL,(R4)+            ; STORE CALL TIMER
 002274                                 RETURN
                                ;
                                ; CLEAR TIMER
                                ;
 002276  116724  000000G        X3CH06: MOVB    $X3CLR,(R4)+            ; STORE CLEAR TIMER
 002302                                 RETURN
                                ;
                                ; RESET TIMER
                                ;
 002304  116724  000000G        X3CH07: MOVB    $X3RES,(R4)+            ; STORE RESET TIMER
 002310                                 RETURN
                                ;
                                ; RESTART TIMER
                                ;
 002312  116724  000000G        X3CH08: MOVB    $X3RST,(R4)+            ; STORE RESTART TIMER
 002316                                 RETURN
                                ;
                                ; MAXIMUM CLEAR VALUE
                                ;
 002320  116724  000000G        X3CH09: MOVB    $X3MCL,(R4)+            ; STORE MAXIMUM CLEAR VALUE
 002324                                 RETURN
                                ;
                                ; MAXIMUM RESETS VALUE
                                ;
 002326  116724  000000G        X3CH10: MOVB    $X3MRE,(R4)+            ; STORE MAXIMUM RESETS VALUE
 002332                                 RETURN
                                ;
                                ; MAXIMUM RESTARTS VALUE
                                ;
 002334  116724  000000G        X3CH11: MOVB    $X3MRS,(R4)+            ; STORE MAXIMUM RESTARTS VALUE
 002340                                 RETURN
```

```
    113 000110                                          TTAB$    PECHA,NULL
    114 000112                                          TTAB$    X3CH01,NULL
    115 000114                                          TTAB$    X3CH02,NULL
    116 000116                                          TTAB$    X3CH03,NULL
    117 000120                                          TTAB$    X3CH04,NULL
    118 000122                                          TTAB$    X3CH05,NULL
    119 000124                                          TTAB$    X3CH06,NULL
    120 000126                                          TTAB$    X3CH07,NULL
    121 000130                                          TTAB$    X3CH08,NULL
    122 000132                                          TTAB$    X3CH09,NULL
    123 000134                                          TTAB$    X3CH10,NULL
    124 000136                                          TTAB$    X3CH11,NULL
    125 000140                                          TTAB$    X2CH01,NULL
    126 000142                                          TTAB$    X2CH02,NULL
    127 000144                                          TTAB$    X2CH03,NULL
    128 000146                                          TTAB$    X2CH04,NULL
    129 000150                                          TTAB$    X2CH05,NULL
    130
    131 000000                                          .PSECT
```

```
000074                                  TTAB$   WORD,POP2
000076                                  TTAB$   APR,POP1
000100                                  TTAB$   DPRB,POP1
000102                                  TTAB$   DPRW,POP2
000104                                  TTAB$   TIME,POP2
000106                                  TTAB$   MXPTB,XMXPTB
000110                                  TTAB$   PECHA,NULL
000112                                  TTAB$   X3CH01,NULL
000114                                  TTAB$   X3CH02,NULL
000116                                  TTAB$   X3CH03,NULL
000120                                  TTAB$   X3CH04,NULL
000122                                  TTAB$   X3CH05,NULL
000124                                  TTAB$   X3CH06,NULL
000126                                  TTAB$   X3CH07,NULL
000130                                  TTAB$   X3CH08,NULL
000132                                  TTAB$   X3CH09,NULL
000134                                  TTAB$   X3CH10,NULL
000136                                  TTAB$   X3CH11,NULL
000140                                  TTAB$   X2CH01,NULL
000142                                  TTAB$   X2CH02,NULL
000144                                  TTAB$   X2CH03,NULL
000146                                  TTAB$   X2CH04,NULL
000150                                  TTAB$   X2CH05,NULL

                                ;
                                ; SECONDARY FILE INDEX VALUE
                                ;
000152                          INDEX:  .BLKW   1

                                ;
                                ; MUX/MTP FLAGS
                                ;
000154                          FLAGS:  .BLKW   1               ; ZERO = NORMAL, NON-ZERO = MUX

                                ;
                                ; DATA NEEDED FOR SEARCH OF UNLOAD BLOCKS
                                ;
000156                          UBIAS:  .BLKW   1               ; BIAS/ADDRESS OF LIBRARY DESCRIPTOR
000160                          UBA:    .BLKW   1               ; OFFSET TO LIBRARY DESCRIPTOR
000162                          CTRADD: .BLKW   1               ; CONTROLLER TABLE STARTING ADDRESS
000164                          BLOCK:  .BLKW   32.             ; LOCAL COPY OF UNLOAD BLOCK (FROM POOL..)

000000                                  .PSECT
```

```
                                         .SBTTL  X2CH
                                 ;
                                 ; MAXIMUM WINDOW SIZE
                                 ;
001750  005203                   X2CH01: INC     R3
001752                                   RETURN
                                 ;
                                 ; MAXIMUM BLOCK SIZE
                                 ;
001754  005723                   X2CH02: TST     (R3)+
001756                                   RETURN
                                 ;
                                 ; MAXIMUM RETRANSMIT COUNT
                                 ;
001760  005203                   X2CH03: INC     R3
001762                                   RETURN
                                 ;
                                 ; RETRANSMIT TIMER
                                 ;
001764  005723                   X2CH04: TST     (R3)+
001766                                   RETURN
                                 ;
                                 ; HOLDBACK TIMER
                                 ;
001770  005723                   X2CH05: TST     (R3)+
001772                                   RETURN


        000001                           .END
```

```
                                         .SBTTL  LOOK FOR DTE$DF MACRO

                               ;+
                               ; $QDTE - LOOK FOR DTE$DF MACRO
                               ;
                               ; INPUTS:
                               ;       NONE
                               ;
                               ; OUTPUTS:
                               ;       C-BIT=SUCCESS/FAILURE
                               ;       R3,R4,R5=DESTROYED
                               ;-
                                                                                                  ;[TD001]
000000                         $QDTE::                                                            ;[TD001]
000000  012705  000000'                MOV     #DTEDF,R5         ; STATE TABLE ADDRESS            ;[TD001]
000004  005001                         CLR     R1                ; FULL KEYWORD MATCH LENGTH      ;**-1
000006  012702  000000'                MOV     #DTEKW,R2         ; KEYWORD TABLE ADDRESS
000012  016703  000000G                MOV     CFGSZ,R3          ; RECORD LENGTH
000016  012704  000000G                MOV     #CFGBF,R4         ; RECORD BUFFER ADDRESS
000022  005067  000000G                CLR     SYNERR            ; CLEAR SYNTAX ERROR FLAG
000026                                 CALL    .TPARS            ; GO DO THE PARSE
000032  103003                         BCC     20$               ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000034  005367  000000G                DEC     SYNERR            ; DID SYNTAX ERROR OCCUR?
000040  001401                         BEQ     101$              ; IF EQ, YES
000042                         20$:    RETURN

                               ;
                               ; ERROR CONDITION
                               ;
000044                         101$:   EMSG$R  1T                ; SYNTAX ERROR
```

```
                                                                                                              N 12
VFGDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:28  Page 20-2
Symbol table

$GETAD= ****** GX        $PSIPT= ****** GX        $SAVAL= ****** GX        $$$KEY= 000002          .PNUMB= ****** GX
$GETRV= ****** GX        $QDTE   000000RG         $SAVRG= ****** GX        $$$R   = 000010          .PNUMH= ****** GX
$LAMDA= 000000           $RADDR= ****** GX        $STRNG= 000004           $$$STA= 000000          .PSTCN= ****** GX
$NUMBR= 000002           $RAD50= 000016           $SUBXP= 000010           $$$TMP= 000012R     005 .PSTPT= ****** GX
$PFLAG= ****** GX        $RBLCK= ****** GX        $$$FLG= 177777           .PCHAR= ****** GX       .TPARS= ****** GX


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001036    001   (RW,I,LCL,REL,CDN)
DATA    000244    002   (RW,D,LCL,REL,CON)
$STATE  000154    003   (RW,D,LCL,REL,CON)
$KTAB   000006    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 16811 Words  ( 66 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:45.06
SY:VFGDTE.V2,[132,134]VFGDTE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFGDTE
```

```
                                          .SBTTL  LOOK FOR PVC$DF MACRO

                                  ;+
                                  ; $QPVC - LOOK FOR PVC$DF MACRO
                                  ;
                                  ; INPUTS:
                                  ;       NONE
                                  ;
                                  ; OUTPUTS:
                                  ;       C-BIT=SUCCESS/FAILURE
                                  ;       R3,R4,R5=DESTROYED
                                  ;-
000000  005067  000032'           $QPVC:: CLR     FLAG            ; CLEAR LOCAL FLAGS WORD
000004  012705  000000'                   MOV     #PVCDF,R5       ; STATE TABLE ADDRESS
000010  005001                            CLR     R1              ; FULL KEYWORD MATCH LENGTH
000012  012702  000000'                   MOV     #PCVKW,R2       ; KEYWORD TABLE ADDRESS
000016  016703  000000G                   MOV     CFGSZ,R3        ; RECORD LENGTH
000022  012704  000000G                   MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000026  005067  000000G                   CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000032                                    CALL    .TPARS          ; GO DO THE PARSE
000036  103003                            BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000040  005367  000000G                   DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000044  001401                            BEQ     101$            ; IF EQ, YES
000046                            20$:    RETURN

                                  ;
                                  ; ERROR CONDITION
                                  ;
000050                            101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
    744 002110  010467  000000G           30$:    MOV     R4,T$LIN+$BFR           ; STORE SLN/STA PAIR
    745 002114  110167  000036G           40$:    MOVB    R1,T$SVC+$BFR+2         ; STORE PORT NUMBER
    746 002120  152767  000200   000044G          BISB    #PX$DLM,T$FLAG+$BFR     ; INDICATE DLM PVC
    747 002126  016767  000000'  000030G          MOV     HTIMR,T$T5+$BFR         ; STORE HELLO TIMER
    748 002134  016767  000002'  000032G          MOV     LTIMR,T$T6+$BFR         ; AND LISTEN TIMER
    749 002142                                    PUTAD                           ; REWRITE UPDATED COUNTER BLOCK
    750 002146                            50$:    RESMAP                          ; RESTORE PREVIOUS MAPPING
    751 002152                            60$:    RETURN
    752
    753         000001                            .END
```

```
        .TITLE  VFGSVC - CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  04-MAR-82
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;


```

B	1	
C	1	
D	1	
E	1	
F	1	
G	1	
H	1	
I	1	
J	1	
K	1	
L	1	
M	1	
N	1	
B	2	
C	2	
D	2	
E	2	
F	2	
G	2	
H	2	
I	2	
J	2	
K	2	
L	2	
M	2	
N	2	
B	3	
C	3	
D	3	
E	3	
F	3	
G	3	
H	3	
I	3	
J	3	
K	3	
L	3	
M	3	
N	3	
B	4	
C	4	
D	4	
E	4	
F	4	
G	4	
H	4	
I	4	
J	4	
K	4	
L	4	
M	4	
N	4	
B	5	
C	5	
D	5	
E	5	
F	5	
G	5	
H	5	
I	5	
J	5	
K	5	
L	5	
M	5	
N	5	VEXP
B	6	
C	6	
D	6	
E	6	
F	6	
G	6	VEX2
H	6	
I	6	
J	6	
K	6	
L	6	
M	6	
N	6	
B	7	
C	7	
D	7	
E	7	
F	7	
G	7	
H	7	
I	7	
J	7	
K	7	
L	7	
M	7	
N	7	
B	8	
C	8	
D	8	
E	8	
F	8	
G	8	
H	8	
I	8	
J	8	VEX3
K	8	
L	8	
M	8	
N	8	
B	9	
C	9	
D	9	
E	9	
F	9	
G	9	
H	9	
I	9	
J	9	VEX5
K	9	
L	9	
M	9	
N	9	
B	10	
C	10	
D	10	
E	10	
F	10	
G	10	
H	10	
I	10	
J	10	
K	10	
L	10	
M	10	
N	10	
B	11	
C	11	
D	11	
E	11	
F	11	
G	11	VFGDTE
H	11	
I	11	
J	11	
K	11	
L	11	
M	11	
N	11	
B	12	
C	12	
D	12	
E	12	
F	12	
G	12	
H	12	
I	12	
J	12	
K	12	
L	12	
M	12	
N	12	
B	13	
C	13	
D	13	
E	13	
F	13	
G	13	VFGPVC
H	13	
I	13	
J	13	
K	13	
L	13	
M	13	
N	13	
B	14	
C	14	
D	14	
E	14	
F	14	
G	14	
H	14	
I	14	
J	14	
K	14	
L	14	
M	14	
N	14	
B	15	
C	15	
D	15	
E	15	
F	15	
G	15	
H	15	
I	15	
J	15	
K	15	
L	15	VFGSVC
M	15	
N	15	
B	16	
C	16	
D	16	
E	16	
F	16	
G	16	
H	16	
I	16	
J	16	
K	16	
L	16	
M	16	

```
                                                .SBTTL  SETTLC - SET UP TRANSPORT LINE COUNTERS

                                        ;+
                                        ;
                                        ; SETTLC - SET UP TRANSPORT LINE COUNTERS
                                        ;
                                        ; INPUTS:
                                        ;       R0 - APR BIAS OF SVC DESCRIPTOR BLOCK
                                        ;       R1 - VIRTUAL ADDRESS OF SVC DESCRIPTOR BLOCK
                                        ;
                                        ; OUTPUTS:
                                        ;       APR BIAS AND VIRTUAL ADDRESS OF SVC DESCRIPTOR BLOCK
                                        ;       STORED IN TRANSPORT LINE COUNTER BLOCK
                                        ;
                                        ;       R0,R1,R2,R3,R4 DESTROYED
                                        ;
                                        ;       NOTE: THIS ROUTINE MUST BE CALLED FROM SYSTEM STATE
                                        ;-
001272  016703  000000G                 SETTLC: MOV     $SLTMA,R3               ; GET SLT VECTOR TABLE ADDRESS
001276  022367  000016'                 5$:     CMP     (R3)+,SLTA              ; IS THIS THE CORRECT ENTRY?
001302  001375                                  BNE     5$                      ; BR IF NO
001304  005743                                  TST     -(R3)                   ; POINT TO CORRECT ENTRY
001306  166703  000000G                         SUB     $SLTMA,R3               ; GET SYSTEM LINE NUMBER*2
001312  066703  000000G                         ADD     $LLCTA,R3               ; INDEX INTO REVERSE MAPPING TABLE
001316  011303                                  MOV     (R3),R3                 ; GET CHANNEL NUMBER AND PDV INDEX
001320  100004                                  BPL     10$                     ; BR IF NOT MULTIPOINT
001322  066703  000014'                         ADD     STASVC,R3               ; POINT TO CORRECT STATION ENTRY
001326  006303                                  ASL     R3                      ; GET ADDRESS OF STATION TABLE
001330  011303                                  MOV     (R3),R3                 ; GET CHANNEL NUMBER AND PDV INDEX
001332  005046                          10$:    CLR     -(SP)                   ; GET INDEX INTO FORWARD MAPPING TABLE
001334  110316                                  MOVB    R3,(SP)                 ; ...
001336  006316                                  ASL     (SP)                    ; ...
001340  012603                                  MOV     (SP)+,R3                ; RETRIEVE FORWARD MAPPING TABLE INDEX
001342  005046                                  CLR     -(SP)                   ; GET XPT'S PDV INDEX
001344  116716  000020'                         MOVB    OWNSVC,(SP)             ; ...
001350  066716  000000G                         ADD     $PDVTA,(SP)             ; POINT TO XPT'S PDV ADDRESS
001354  013602                                  MOV     @(SP)+,R2               ; GET XPT'S PDV ADDRESS
001356  060203                                  ADD     R2,R3                   ; POINT TO FORWARD MAPPING TABLE ENTRY
001360  062703  000000G                         ADD     #Z.MAP,R3               ; ...
001364  011304                                  MOV     (R3),R4                 ; GET SLN/STA PAIR
001366                                          SAVMAP                          ; SAVE CURRENT MAPPING
001372                                          GETRV   Z.DAT(R2),#N$XLEN       ; READ IN XPT'S DDB
001412                                          MAP     N$TLC+2+$BFR            ; MAP TO COUNTERS
001420  016703  000100G                         MOV     N$TLC+$BFR,R3           ; GET NUMBER OF LINE COUNTERS
001424  016702  000104G                         MOV     N$TLC+4+$BFR,R2         ; GET ADDRESS OF FIRST COUNTER BLOCK

001430                                  20$:    GETAD   R2,#T$LLEN              ; READ IN COUNTER BLOCK
001446  005767  000000G                         TST     T$LIN+$BFR              ; HAS THIS BLOCK BEEN ASSIGNED ALREADY
001452  100410                                  BMI     30$                     ; BR IF NO - USE IT
001454  026704  000000G                         CMP     T$LIN+$BFR,R4           ; ELSE IS THIS THE ONE WE WANT?
001460  001407                                  BEQ     40$                     ; BR IF YES
001462  062702  000046                          ADD     #T$LLEN,R2              ; POINT TO NEXT COUNTER BLOCK
001466                                          SOB     R3,20$                  ; CHECK ALL COUNTER BLOCKS
001472  000432                                  BR      60$                     ; CONTINUE
001474  010467  000000G                 30$:    MOV     R4,T$LIN+$BFR           ; STORE SLN/STA PAIR
001500  152767  000300  000044G         40$:    BISB    #PX$SVC!PX$DLM,T$FLAG+$BFR ; INDICATE DLM SVC
001506  005067  000034G                         CLR     T$SVC+$BFR              ; ASSUME SVC DESCRIPTOR IS IN DSR
```

Table of contents

```
     5-    55     MACRO DEFINITIONS
     6-    72     CONSTANT DEFINITIONS
     7-    82     LOCAL DATA
     8-    92     ERROR MESSAGES
     9-   103     $QX2P - LOOK FOR X2P$DF MACRO
    10-   132     X2P$DF STATE TABLE
    11-   195     X2P$DF ACTION ROUTINES
```

```
                                                                       B 3
VFGX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32       PAGE 1
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE               REFERENCES

CERR     = ******  GX        8-98         8-99         8-100
CFGBF    = ******  GX        9-120
CFGSZ    = ******  GX        9-119
CFLIN    = ******  GX        8-98         8-99         8-100
CHKDEV     000316 R         #11-225
FMT8B    = ******  GX        8-98         8-99         8-100
FM.8B    = 000000           #8-98        #8-99        #8-100
I$$AS    = ******            5-62
LF.ACT   = 100000           #5-63
LF.BRO   = 000400           #5-63
LF.BWT   = 000007           #5-63
LF.ENA   = 002000           #5-63
LF.LPB   = 001000           #5-63
LF.MDC   = 000100           #5-63
LF.MFL   = 004000           #5-63
LF.MTP   = 000020           #5-63
LF.PAC   = 000200           #5-63
LF.RDY   = 040000           #5-63
LF.REA   = 010000           #5-63
LF.SER   = 000040           #5-63
LF.TIM   = 000010           #5-63
LF.UNL   = 020000           #5-63
LF.X2P   = 000000           #5-63
LN.CLO   = 000000           #5-63
LN.DUM   = 000005           #5-63
LN.LOA   = 000004           #5-63
LN.LOO   = 000003           #5-63
LN.OAU   = 000003           #5-63
LN.OFF   = 000001           #5-63
LN.ON    = 000000           #5-63
LN.OOP   = 000004           #5-63
LN.OPE   = 000001           #5-63
LN.REF   = 000002           #5-63
LN.SER   = 000002           #5-63
LN.STA   = 000017           #5-63
LN.SUB   = 000360           #5-63
LN.TRI   = 000006           #5-63
L.COST     000015           #5-63
L.CTL      000012           #5-63         11-231
L.CVA      177776           #5-63
L.DDM      000002           #5-63         11-226
L.DDS      000004           #5-63
L.DLC      000003           #5-63
L.DLM      000006           #5-63
L.DLS      000010           #5-63
L.FLG      000000           #5-63
L.KRBA     000016           #5-63
L.LEN    = 000022           #5-63
L.MPF      000022           #5-63
L.NMST     000020           #5-63
L.NSTA     000014           #5-63
L.OWNR     000021           #5-63
```

```
                                                                       C 3
VFGX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32       PAGE 2
```

```
                                      .SBTTL  X3P$DF STATE TABLE
                                ;
                                ; TPARS STATE TABLES
                                ;
 000052                               ISTAT$  X3PST,X3PKW
                                ;
                                ; X3P$DF
                                ;
 000052                               STATE$  X3PDF
 000052                                TRAN$   %X3P$DF%,,,1,SYNERR

 000052                               STATE$
 000052                                TRAN$   $NUMBR,,X3.DBS         ; DEFAULT BLOCK SIZE

 000052                               STATE$
 000052                                TRAN$   <',>

 000052                               STATE$
 000052                                TRAN$   $NUMBR,,X3.MBS         ; MAXIMUM BLOCK SIZE

 000052                               STATE$
 000052                                TRAN$   <',>

 000052                               STATE$
 000052                                TRAN$   $NUMBR,,X3.DWS         ; DEFAULT WINDOW SIZE

 000052                               STATE$
 000052                                TRAN$   <',>

 000052                               STATE$
 000052                                TRAN$   $NUMBR,,X3.MWS         ; MAXIMUM WINDOW SIZE

 000052                               STATE$
 000052                                TRAN$   <',>

 000052                               STATE$
 000052                                TRAN$   $NUMBR,,X3.BEG         ; BEGINNING OF TIMER VALUES

 000052                               STATE$  X3NXT
 000052                                TRAN$   $EOS,$EXIT,X3.END
 000052                                TRAN$   <',>

 000052                               STATE$
 000052                                TRAN$   $NUMBR,X3NXT,X3.NXT    ; NEXT TIMER VALUE

 000052                               STATE$
```

```
                                                .SBTTL  $FNALI - FIND ALIAS
                                        ;+
                                        ; *** - $FNALI - FIND ALIAS
                                        ;
                                        ; INPUTS:
                                        ;       R1 = ADDRESS OF ALIAS NAME STRING
                                        ;       RQB+LR.UCB = UCB ADDRESS OF DEVICE SCOPE
                                        ;       $QUAL = QUALIFIERS FOR KNOWN OR ALL ALIASES
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       R0 = UNMAPPED ADDRESS OF ALIAS BLOCK, IF FOUND
                                        ;       R1 = UNMAPPED ADDRESS OF PREVIOUS BLOCK
                                        ;-

 000116                                 $FNALI::
 000116  010246                                 MOV     R2,-(SP)                ; SAVE R2

 000120                                         SWTCHI  #$HMBUF                 ; USE TEMPORARY BUFFER
 000126                                         GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
 000146                                         SWTCHI                          ; RESET DEFAULT BUFFER

 000154  012700  000000G                        MOV     #$HMBUF+D$ANN,R0        ; GET ALIAS LISTHEAD ADDRESS

 000160  011067  000000G                        MOV     (R0),$BFR               ; GET FIRST ALIAS IN LIST
 000164  010146                                 MOV     R1,-(SP)                ; SAVE ADDRESS OF ALIAS NAME
 000166  010067  177606                         MOV     R0,CURUNM               ; SAVE UNMAPPED ADDR OF CURRENT BLOCK
 000172  016701  177602                 10$:    MOV     CURUNM,R1               ; SAVE ADDRESS OF PREVIOUS
 000176  016767  000000G 177574                 MOV     $BFR,CURUNM             ; GET NEXT OBJECT IN LIST
 000204  000261                                 SEC                             ; ASSUME FAILURE
 000206  001445                                 BEQ     50$                     ; IF EQ, NO MORE
 000210                                         CEACC$  CURUNM,R0               ; CONVERT TO MAPPED ADDRESS
 000222                                 20$:    GETAD   R0,#100                 ; READ ALIAS IN INTERNAL BUFFER
 000240  032767  000002  000000G                BIT     #QF$ALL,$QUAL           ; IS THIS FOR ALL ALIASES ?
 000246  001022                                 BNE     40$                     ; IF NE, YES .. SUCCESS
 000250  032767  000004  000000G                BIT     #QF$KNO,$QUAL           ; IS THIS FOR KNOWN ALIASES ?
 000256  001012                                 BNE     30$                     ; IF NE, YES .. JUST CHECK SCOPE
 000260  011602                                 MOV     (SP),R2                 ; GET ADDRESS OF ALIAS STRING
 000262  022267  000002G                        CMP     (R2)+,$BFR+A.NAM        ; IS THIS THE RIGHT ALIAS ?
 000266  001341                                 BNE     10$                     ; IF NE, NO
 000270  022267  000004G                        CMP     (R2)+,$BFR+A.NAM+2      ; IS THIS REALLY THE ALIAS ?
 000274  001336                                 BNE     10$                     ; IF NE, NO
 000276  022267  000006G                        CMP     (R2)+,$BFR+A.NAM+4      ; THIS IS THE LAST CHANCE
 000302  001333                                 BNE     10$                     ; IF NE .. SO SORRY
 000304  026767  000016G 000010G        30$:    CMP     RQB+LR.UCB,$BFR+A.UCB   ; BUT IS THIS THE RIGHT SCOPE ?
 000312  001327                                 BNE     10$                     ; IF NE, NO .. TOO BAD
 000314  016700  177460                 40$:    MOV     CURUNM,R0               ; GET UNMAPPED ADDRESS OF BLOCK
 000320  000241                                 CLC                             ; ELSE, INDICATE SUCCESS
 000322  005226                         50$:    INC     (SP)+                   ; CLEAN UP STACK
 000324  012602                                 MOV     (SP)+,R2                ; RESTORE R2
 000326                                         RETURN
```

```
        .TITLE  VFPCUG - INITIAL CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/
        .ENABL  LC                                                        ;[TD001]
                                                                          ;**-1
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; NTL - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-82
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                          .SBTTL  ALCUG1 - Allocate a CUG name block (old format)                  ;[TD001]
                                 ;+                                                                                ;[TD001]
                                 ; *** ALCUG1 - Allocate a CUG name block (old format)                             ;[TD001]
                                 ;                                                                                 ;[TD001]
                                 ; INPUTS                                                                          ;[TD001]
                                 ;        All parsed fields of CUG$DF macro                                        ;[TD001]
                                 ;                                                                                 ;[TD001]
                                 ; OUTPUTS                                                                         ;[TD001]
                                 ;        Carry clear - CUG name block allocated and linked into CUG chain         ;[TD001]
                                 ;        Carry set - Insufficient store to allocate CUG name block                ;[TD001]
                                 ;-                                                                                ;[TD001]
000672                           ALCUG1:                                                                           ;[TD001]
000672  005000                            CLR     R0                      ; Nowhere to store name length           ;[TD001]
000674  012701  000002                    MOV     #G$NAM,R1               ; Place to store name                    ;[TD001]
000700  012702  000016                    MOV     #G$LEN,R2               ; Length to allocate                     ;[TD001]
000704                                    CALL    ALCUG                   ; allocate and fill block                ;[TD001]
000710                                    RETURN                                                                   ;[TD001]
```

```
                                        .SBTTL  LOOK FOR DSA$DF MACRO

                                ;+
                                ; $QDSA - LOOK FOR DSA$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3,R4 R5=DESTROYED
                                ;-
                                        .ENABL  LSB

000000  012705  000000'         $QDSA:: MOV     #DSADF,R5       ; STATE TABLE ADDRESS                           ;[TD001]
000004  000402                          BR      10$                                                             ;[TD001]
000006                          $QDSC::                                                                         ;[TD001]
000006  012705  000130'                 MOV     #DSCDF,R5       ; STATE TABLE ADDRESS                           ;[TD001]
000012                          10$:    VNPDBG  VFPDSA          ; VNP debugging break point                     ;[TD001]
000012  005001                          CLR     R1              ; FULL KEYWORD MATCH LENGTH                     ;[TD001]
000014  012702  000000'                 MOV     #DSAKW,R2       ; KEYWORD TABLE ADDRESS                         ;**-4
000020  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
000024  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000030  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000034                                  CALL    .TPARS          ; GO DO THE PARSE
000040  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000042  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000046  001401                          BEQ     101$            ; IF EQ, YES
000050                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000052                          101$:   EMSG$R  1T              ; SYNTAX ERROR
                                        .DSABL  LSB
```

B 9

```
VFPDSN - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:34
Table of contents
```

```
     5-   50    Local Symbols
     6-   65    Macro definitions
     7-   81    Local Data
     8-   90    Look for DSN$DF macro
     9-  120    Parse table for DSN$DF
    11-  213    Syntax action routines
    15-  309    Parser Subexpression Action Routines
```

C 9

```
                                        ;+
                                        ; *** STORCH - Store string character
                                        ;
                                        ; INPUTS
                                        ;       STRLEN - Number of characters read so far
                                        ;       .PCHAR - Current .haracter to be stored
                                        ;
                                        ; OUTPUTS
                                        ;       STRLEN incremented
                                        ;       If chars read <= max string size, character stored in string buffer
                                        ;-
000272  126727  000040' 000020  STORCH: CMPB    STRLEN,#MAXSTR          ; Check string not too long
000300  103005                          BHIS    10$                     ; Do not store character if no room
000302  116700  000040'                 MOVB    STRLEN,R0
000306  116760  000000G 000041'         MOVB    .PCHAR,STRBUF(R0)       ; Store character in buffer
000314  105267  000040'         10$:    INCB    STRLEN                  ; Count this character
000320                                  RETURN
        000001                          .END
```

```
                                        .SBTTL  DST$DF ACTION ROUTINES

                                ;
                                ; DESTINATION NAME
                                ;
000056  012700  000000G         DSNAM:  MOV     #DSTNML,R0              ; Buffer for destination name           ;[TD001]
000062  012701  000006                  MOV     #DSNAMX,R1              ; Max length if old-format              ;[TD001]
000066                                  CALL    STORID                  ; Check and store destination name      ;[TD001]
000072  103430                          BCS     101$                    ; Branch if bad name                    ;[TD001]
000074  105067  000000G                 CLRB    DSACGL                  ; Clear CUG name length                 ;[TD001]
000100  105067  000000G                 CLRB    DSTTKL                  ; Assume no task name                   ;[TD001]
000104  105067  000000G                 CLRB    RNNL                    ; Clear remote node name                ;[TD001]
000110  105067  000000G                 CLRB    RNUL                    ; Clear remote user name                ;[TD001]
000114  105067  000000G                 CLRB    RNPL                    ; Clear remote password                 ;[TD001]
000120  105067  000000G                 CLRB    RNAL                    ; Clear remote account                  ;[TD001]
000124  005067  000000G                 CLR     DSASLO                  ; DEFAULT LOW SUBBADDRESS TO ZERO       ; *-21
000130  012767  177777  000000G         MOV     #-1,DSASHI              ; ASSUME NO SUBADDRESS RANGE
000136  105067  000000G                 CLRB    DSCMCT                  ; CLEAR CALL DATA MASK COUNT
000142  105067  000000G                 CLRB    DSCVCT                  ; CLEAR CALL DATA VALUE COUNT
000146  105067  000000G                 CLRB    DSARCT                  ; CLEAR REMOTE DTE ADDRESS COUNT
000152                                  RETURN
                                ;
                                ; ERRORS
                                ;
000154                          101$:   EMSG$R  N4                      ; ILLEGAL DESTINATION NAME
```

```
                                .SBTTL  MOVAR - Move variable-length byte fields                ;[TD001]
                        ;+                                                                      ;[TD001]
                        ; *** MOVAR - Move variable-length byte fields                          ;[TD001]
                        ;                                                                       ;[TD001]
                        ; FUNCTION                                                              ;[TD001]
                        ;       Given table of pointers to variable length fields               ;[TD001]
                        ;       (data with one-byte precount), move all fields                  ;[TD001]
                        ;       contiguously to output buffer                                   ;[TD001]
                        ;                                                                       ;[TD001]
                        ; INPUTS                                                                ;[TD001]
                        ;       R0 - Address of table of pointers to fields                     ;[TD001]
                        ;       R1 - Address of output buffer                                   ;[TD001]
                        ;                                                                       ;[TD001]
                        ; OUTPUTS                                                               ;[TD001]
                        ;       R1 - Next free byte in output buffer                            ;[TD001]
                        ;       R0, R2, R3 destroyed                                            ;[TD001]
                        ;-                                                                      ;[TD001]
001662                  MOVAR:                                                                  ;[TD001]
001662  012002          5$:     MOV     (R0)+,R2                ; Get next table entry          ;[TD001]
001664  001406                  BEQ     20$                     ; Zero at end of table          ;[TD001]
001666  112203                  MOVB    (R2)+,R3                ; Get length of entry           ;[TD001]
001670  001774                  BEQ     5$                      ; If empty, look for next       ;[TD001]
001672  112221          10$:    MOVB    (R2)+,(R1)+             ; Move data to output area      ;[TD001]
001674  005303                  DEC     R3                                                      ;[TD001]
001676  003375                  BGT     10$                                                     ;[TD001]
001700  000770                  BR      5$                      ; loop                          ;[TD001]
                        ;                                                                       ;[TD001]
001702                  20$:    RETURN                                                          ;[TD001]
```

```
     81                                          .SBTTL  LOCAL SYMBOL DEFINITIONS
     82
     83                                  ;
     84                                  ; LOCAL SYMBOL DEFINITIONS
     85                                  ;
     86          000040                          SPACE = 40              ; ASCII SPACE
     87          000040                          HSHMN  = 32.            ; MINIMUM VALUE FOR HASH TABLE SIZE
     88          001000                          HSHMX  = 512.           ; MAXIMUM VALUE FOR HASH TABLE SIZE
     89          000020                          NUMDTE = 16.            ; MAXIMUM DTES ALLOWED
     90
     91                                  ;
     92                                  ; SAVED REGISTER OFFSETS ON STACK FOR $SAVAL
     93                                  ;
     94          000002                          R$R0 = 2                ; SAVED R0
     95          000004                          R$R1 = 4                ; SAVED R1
     96          000006                          R$R2 = 6                ; SAVED R2
     97          000010                          R$R3 = 10               ; SAVED R3
     98          000012                          R$R4 = 12               ; SAVED R4
     99          000014                          R$R5 = 14               ; SAVED R5
```

```
                                                .SBTTL  BIASX - SET APR6 BIAS CORRECTLY

                                        ;+
                                        ;
                                        ; BIASX - SET APR6 BIAS CORRECTLY
                                        ;
                                        ; INPUTS:
                                        ;       R0 = ADDRESS (POOL OR APR5)
                                        ;
                                        ; OUTPUTS:
                                        ;       R0 = ADDRESS (POOL OR APR6)
                                        ;
                                        ;-

 001322                                 BIASX:
 001322  020027  120000                         CMP     R0,#120000              ;; ADDRESS IN APR5 ?
 001326  103404                                 BLO     10$                     ;; BR IF NO (ASSUME POOL)
 001330  042700  160000                         BIC     #160000,R0              ;; CLEAR ALL APR BITS
 001334  052700  140000                         BIS     #140000,R0              ;; AND BIAS FOR APR6
 001340                                 10$:    RETURN                          ;; RETURN

         000001                                 .END
```

```
000000                               .PSECT  DATA,D                                                       ;**-20

                                     .NLIST  BEX
                              ;
                              ; LOCAL DATA FOR PSN$DF MACRO
                              ;
000000                        PSNFLG: .BLKW   1               ; FLAGS WORD FROM PSN$DF MACRO              ;**-1
000002                        PSNLO:  .BLKW   1               ; LOW END OF SUBADDRESS RANGE
000004                        PSNHI:  .BLKW   1               ; HIGH END OF SUBADDRESS RANGE
000006                        PSNNML: .BLKB   1               ; Length of PSN network name                ;[TD001]
000007                        PSNNAM: .BLKB   MAXID           ; PSN network name from PSN$DF macro        ;[TD001]
                                      .EVEN                                                              ;[TD001]
```

```
VFPPSN      CREATED BY  MACRO  ON 29-JUN-85 AT 02:37      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

PSVTSZ  = 000400            #6-87      15-422     15-426     15-432     15-434     15-437
RTSPC   = ******   GX        8-105     8-106      8-107      8-108      8-109      8-110      8-111      8-112      8-113
SPACE   = 000040            #6-88
SPFILL  = ******   GX        11-212
SUBAMX  = ******   GX        12-291
SYNERR  = ******   GX       *9-128     *9-141
SYSFDB  = ******   GX       *13 343    *14-398    *14-403    *15-432    *15-437    *15-439
S$$BAS  = ******             8-105     8-105      8-106      8-106      8-107      8-107      8-108      8-108      8-109
                             8-109     8-110      8-110      8-111      8-111      8-112      8-112      8-113      8-113
$ALL16  = ******   GX        13-320
$ALPHA  = 000022            #10-154
$ALPOL  = ******   GX        14-386    15-427
$ANY    = 000020            #10-154
$BFR    = ******   GX        13-323    13-329     13-333     *13-337    *13-338    *13-340    *13-341    14-391     *14-402
                             15-421    *15-440    *15-441
$BLANK  = 000006            #10-154
$CEACC  = ******   GX        14-390    15-431     15-436
$DBG$   = ******             11-206
$DIGIT  = 000024            #10-154
$DNUMB  = 000014            #10-154
$EOS    = 000012            #10-154
$ERRD1    000366 R          #8-112     12-282
$ERRD2    000424 R          #8-113     13-362
$ERRPA    000310 R          #8-111     13-365
$ERRP1    000030 R          #8-105     11-262
$ERRP2    000066 R          #8-106     11-263
$ERRP3    000114 R          #8-107     11-264
$ERRP4    000144 R          #8-108     11-265
$ERRP5    000170 R          #8-109     13-363
$ERRP6    000242 R          #8-110     13-364
$ERR1T  = ******   GX        9-148
$EXIT   = 000000            #10-154
$FAIL   = 177777            #10-154
$GETRV  = ******   GX        15-439
$GPRM   = ******             10-154
$LAMDA  = 000000            #10-154
$NUMBR  = 000002            #10-154
$PFLAG  = ******   GX       *11-242    *11-245    *11-248    *11-253    11-254     13-313     13-330     13-353
$PSIPT  = ******   GX        11-207    *13-344    14-400     15-439
$PUTAD  = ******   GX        14-398    15-432     15-437
$PUTRC  = ******   GX        13-343    14-403     15-442
$QPSN     000000 RG         #9-132
$RADDR  = ******   GX       *13-343    *14-398    *14-403    *15-432    *15-437    *15-439
$RAD50  = 000016            #10-154
$RBLCK  = ******   GX        14-389    *14-399    15-430     *15-438
$RONLY  = ******             10-154    10-154     10-154
$SAVAL  = ******   GX        13-311
$STRNG  = 000004            #10-154
$SUBXP  = 000010            #10-154
$$$FLG  = 177777            #10-154
$$$KEY  = 177777            #10-154
.PNUMB  = ******   GX        11-230    11-241     12-289
```

```
    584 001512  020127  120000                  CMP     R1,#120000              ; IS SVC DESCRIPTOR IN DSR?
    585 001516  103406                          BLO     50$                     ; BR IF YES
    586 001520  042701  160000                  BIC     #160000,R1              ; CLEAR CURRENT APR MAPPING BITS
    587 001524  052701  140000                  BIS     #140000,R1              ; XPT NEEDS APR6 MAPPING
    588 001530  010067  000034G                 MOV     R0,T$SVC+$BFR           ; STORE APR BIAS OF SVC DESCRIPTOR
    589 001534  010167  000036G         50$:    MOV     R1,T$SVC+$BFR+2         ; STORE VIRTUAL ADDR OF SVC DESCRIPTOR
    590 001540  016767  000002' 000030G         MOV     HTIMR,T$T5+$BFR         ; STORE HELLO TIMER
    591 001546  016767  000004' 000032G         MOV     LTIMR,T$T6+$BFR         ; AND LISTEN TIMER
    592 001554                                  PUTAD                           ; REWRITE UPDATED COUNTER BLOCK
    593 001560                          60$:    RESMAP                          ; RESTORE PREVIOUS MAPPING
    594 001564                                  RETURN
```

```
        .TITLE  VFGX2P - CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-82
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE               REFERENCES

L.UNT      000013            #5-63        11-233
RTSPC    = ******    GX       8-98        8-99          8-100
R$$11D   = ******             5-62
R$$11M   = 000000             5-62
R$$11S   = ******             5-62
SF.ACT   = 000200            #5-63
SF.ENA   = 000100            #5-63
SF.LPB   = 000004            #5-63
SF.MFL   = 000040            #5-63
SF.PAC   = 000020            #5-63
SF.REA   = 000010            #5-63
SF.SER   = 000001            #5-63
SF.SVC   = 000002            #5-63
SF.UNL   = 000040            #5-63
SYNERR   = ******    GX      *9-121      *9-124        *11-235
S$$DAS   = ******             8-98        8-98          8-99          8-99          8-100         8-100
S.COST     000001            #5-63
S.FLG      000000            #5-63
S.LEN      000004            #5-63
S.NMST     000002            #5-63
S.OWNR     000003            #5-63
X$$MCB   = ******             5-62        5-62
X2CTL      000002 R          #7-88       *11-214        11-231
X2DEV      000000 R          #7-87       *11-205        11-229
X2HLMX   = 001130            #6-80        11-291
X2PDF      000000 R           9-116
X2PKW      000000 RG          9-118      #10-136
X2PST      000000 RG         #10-136
X2RTMX   = 001130            #6-79        11-279
X2UNT      000003 R          #7-89       *11-215       *11-224        11-233
X2WNMN   = 000001            #6-78        11-248
X2WNMX   = 000007            #6-77        11-246
X2.CTL     000244 R          #11-210
X2.DEV     000216 R          #11-201
X2.HLD     000564 R          #11-287
X2.MBS     000446 R          #11-256
X2.MRT     000474 R          #11-265
X2.MWS     000404 R          #11-243
X2.RTR     000526 R          #11-275
X2.UNT     000274 R          #11-220
ZF.COU   = 001000            #5-62
ZF.DDM   = 000001            #5-62
ZF.DIA   = 004000            #5-62
ZF.DLC   = 000002            #5-62
ZF.DVP   = 100000            #5-62
ZF.INI   = 040000            #5-62
ZF.KMX   = 000020            #5-62
ZF.LLC   = 000004            #5-62
ZF.LMC   = 000100            #5-62
ZF.MAN   = 020000            #5-62
ZF.MFL   = 000010            #5-62
ZF.MTM   = 000400            #5-62
```

```
176                                             .SBTTL  X3P$DF ACTION ROUTINES
177
178                                     ;
179                                     ; DEFAULT BLOCK SIZE
180                                     ;
181 000052                              X3.DBS: CALL    CHKBLK                  ; CHECK FOR LEGAL DEFAULT BLOCK SIZE
182 000056  103403                              BCS     101$                    ; BR IF ERROR
183 000060  010067  000000G                     MOV     R0,$X3DBS               ; STORE DEFAULT BLOCK SIZE
184 000064                                      RETURN
185 000066                              101$:   EMSG$R  2B                      ; ILLEGAL DEFAULT BLOCK SIZE
186                                     ;
187                                     ; MAXIMUM BLOCK SIZE
188                                     ;
189 000074                              X3.MBS: CALL    CHKBLK                  ; CHECK FOR LEGAL MAXIMUM BLOCK SIZE
190 000100  103406                              BCS     101$                    ; BR IF ILLEGAL VALUE
191 000102  020067  000000G                     CMP     R0,$X3DBS               ; IS DEFAULT IN RANGE?
192 000106  103406                              BLO     102$                    ; BR IF NO
193 000110  010067  000000G                     MOV     R0,$X3MBS               ; SAVE MAXIMUM BLOCK SIZE
194 000114                                      RETURN
195 000116                              101$:   EMSG$R  2C                      ; ILLEGAL MAXIMUM BLOCK SIZE
196 000124                              102$:   EMSG$R  2D                      ; DEFAULT BLOCK SIZE EXCEEDS MAXIMUM
197                                     ;
198                                     ; DEFAULT WINDOW SIZE
199                                     ;
200 000132                              X3.DWS: CALL    CHKWND                  ; CHECK FOR LEGAL WINDOW SIZE
201 000136  103403                              BCS     101$                    ; BR IF ILLEGAL SIZE - ERROR
202 000140  010067  000000G                     MOV     R0,$X3DWS               ; STORE DEFAULT WINDOW SIZE
203 000144                                      RETURN
204 000146                              101$:   EMSG$R  2E                      ; ILLEGAL DEFAULT WINDOW SIZE
205                                     ;
206                                     ; MAXIMUM WINDOW SIZE
207                                     ;
208 000154                              X3.MWS: CALL    CHKWND                  ; CHECK FOR LEGAL WINDOW SIZE
209 000160  103406                              BCS     101$                    ; BR IF ERROR
210 000162  020067  000000G                     CMP     R0,$X3DWS               ; IS DEFAULT IN RANGE?
211 000166  103406                              BLO     102$                    ; BR IF NO - ERROR
212 000170  010067  000000G                     MOV     R0,$X3MWS               ; STORE MAXIMUM WINDOW SIZE
213 000174                                      RETURN
214 000176                              101$:   EMSG$R  2F                      ; ILLEGAL MAXIMUM WINDOW SIZE
215 000204                              102$:   EMSG$R  2G                      ; DEFAULT WINDOW SIZE EXCEEDS MAXIMUM
216                                     ;
217                                     ; BEGINNING OF TIMER VALUES
218                                     ;
219 000212  012767  000000G 000000'     X3.BEG: MOV     #$X3CAL,NEXT            ; POINT TO CALL TIMER VALUE
220 000220  105067  000010'                     CLRB    TIMRCT                  ; CLEAR COUNTER
221                                     ;
222                                     ; NEXT TIMER VALUE
223                                     ;
224 000224  105267  000010'             X3.NXT: INCB    TIMRCT                  ; UPDATE COUNT
225 000230  126727  000010' 000007              CMPB    TIMRCT,#X3TMCT          ; MAXIMUM EXCEEDED?
226 000236  101015                              BHI     101$                    ; BR IF YES - ERROR
227 000240  005767  000000G                     TST     .PNUMH                  ; DOUBLE WORD VALUE?
228 000244  001017                              BNE     102$                    ; BR IF YES - ERROR
229 000246  105767  000001G                     TSTB    .PNUMB+1                ; BYTE VALUE?
230 000252  001014                              BNE     102$                    ; BR IF NO - ERROR
231 000254  016777  000000G 000000'             MOV     .PNUMB,@NEXT            ; STORE TIMER VALUE
232 000262  062767  000002  000000'             ADD     #2,NEXT                 ; POINT TO STORAGE FOR NEXT VALUE
```

```
                                                  .SBTTL  $FNADD - FIND REMOTE BY ADDRESS
                                          ;+
                                          ; *** - $FNADD - FIND A REMOTE NODE BY ADDRESS
                                          ;
                                          ; INPUTS:
                                          ;       R1 = NODE ADDRESS
                                          ;
                                          ; OUTPUTS:
                                          ;       C-BIT = SUCCESS/FAILURE
                                          ;       R0 = UNMAPPED ADDRESS OF REMOTE BLOCK, IF FOUND
                                          ;       R1 = UNMAPPED ADDRESS OF PREVIOUS
                                          ;-

000330                                    $FNADD::
000330                                            SWTCHI  #$HMBUF                 ; USE TEMPORARY BUFFER
000336                                            GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
000356                                            SWTCHI                          ; RESET DEFAULT BUFFER

000364  012700  000004G                           MOV     #$HMBUF+D$RNN+2,R0      ; GET START OF REMOTES LIST

000370  011067  000000G                           MOV     (R0),$BFR               ; GET FIRST IN LIST
000374  010146                                    MOV     R1,-(SP)                ; SAVE NODE ADDRESS ON STACK
000376  010067  177376                            MOV     R0,CURUNM               ; SAVE UNMAPPED ADDR OF CURRENT BLOCK
000402  016701  177372                    10$:    MOV     CURUNM,R1               ; SAVE ADDRESS OF PREVIOUS
000406  016767  000000G 177364                    MOV     $BFR,CURUNM             ; GET NEXT OBJECT IN LIST
000414  000261                                    SEC                             ; ASSUME FAILURE
000416  001422                                    BEQ     30$                     ; IF EQ, NO MORE
000420                                            CEACC$  CURUNM,R0               ; CONVERT TO MAPPED ADDRESS
000432                                    20$:    GETAD   R0,#R.LEN               ; READ REMOTE IN INTERNAL BUFFER
000450  021667  000010G                           CMP     (SP),$BFR+R.ADD         ; IS THIS THE RIGHT REMOTE ?
000454  001352                                    BNE     10$                     ; IF NE, NO
000456  016700  177316                            MOV     CURUNM,R0               ; GET UNMAPPED ADDRESS OF BLOCK
000462  000241                                    CLC                             ; ELSE, INDICATE SUCCESS
000464  005226                            30$:    INC     (SP)+                   ; CLEAN UP THE STACK
000466                                            RETURN
```

```
     56                                                 .SBTTL  MACRO DEFINITIONS
     57
     58                                         ;
     59                                         ; LIBRARY MACROS
     60                                         ;
     61                                                 .MCALL  CUGDF$,EMSG$R,PHBDF$,NTLER$,STATE$,TRAN$,ISTAT$,PUTAD
     62                                                 .MCALL  CEACC$,GETAD,GETRV
     63                                                 .MCALL  VNPDBG                                                  ;[TD001]
     64
     65 000000                                          CUGDF$                          ; CLOSED USER GROUP BLOCK OFFSETS
     66 000000                                          PHBDF$                          ; PSI HOME BLOCK OFFSETS
     67
     68                                         ;
     69                                         ; LOCAL MACRO DEFINITIONS
     70                                         ;
     71                                                 .MACRO  SAVMAP
     72                                                 MOV     $RBLCK,-(SP)
     73                                                 .ENDM
     74
     75                                                 .MACRO  RESMAP
     76                                                 MOV     (SP)+,$RBLCK
     77                                                 .ENDM
     78
```

```
 398                                           .SBTTL  ALCUG - Allocate and fill a CUG name block (general)          ;[TD001]
 399                                   ;+                                                                               ;[TD001]
 400                                   ; *** ALCUG - Allocate and fill a CUG name block (general)                       ;[TD001]
 401                                   ;                                                                                ;[TD001]
 402                                   ; INPUTS                                                                         ;[TD001]
 403                                   ;       R0      - Offset in CUG name block to store CUG name length (0 if none)  ;[TD001]
 404                                   ;       R1      - Offset in CUG name block to store CUG name                     ;[TD001]
 405                                   ;       R2      - Total length of CUG name block (bytes)                         ;[TD001]
 406                                   ;       All parsed fields of CUG$DF macro                                        ;[TD001]
 407                                   ;                                                                                ;[TD001]
 408                                   ; OUTPUTS                                                                        ;[TD001]
 409                                   ;       Carry set if insufficient store to allocate block                        ;[TD001]
 410                                   ;       All registers preserved                                                  ;[TD001]
 411                                   ;-                                                                               ;[TD001]
 412                                           .ENABL  LSB                                                              ;[TD001]
 413 000712                            ALCUG:                                                                           ;[TD001]
 414 000712                                    CALL    $SAVAL                  ; Save registers                         ;[TD001]
 415 000716                                    SAVMAP                          ; Save current mapping                   ;[TD001]
 416 000722 010003                             MOV     R0,R3                   ; Keep a copy of input parameters        ;[TD001]
 417 000724 010104                             MOV     R1,R4                                                            ;[TD001]
 418 000726 010201                             MOV     R2,R1                   ; Get length to allocate                 ;[TD001]
 419 000730                                    CALL    $ALPOL                  ; Attempt to allocate block              ;[TD001]
 420 000734 103465                             BCS     99$                     ; Branch if allocation fails             ;[TD001]
 421 000736 010005                             MOV     R0,R5                   ; Save unmapped address of allocation    ;[TD001]
 422 000740                                    CEACC$  R0                      ; Map to allocation                      ;[TD001]
 423 000750                                    GETAD   R0,R1                   ; Get allocated block                    ;[TD001]
 424 000764 012702  000000G                    MOV     #$BFR,R2                ; Clear allocated block                  ;[TD001]
 425 000770 105022                     20$:    CLRB    (R2)+                                                            ;[TD001]
 426 000772 005301                             DEC     R1                                                               ;[TD001]
 427 000774 003375                             BGT     20$                                                              ;[TD001]
 428                                   ;                                                                                ;[TD001]
 429 000776 012701  000002                     MOV     #<CUGNLN/2>,R1          ; Insert CUG number                      ;[TD001]
 430 001002 012702  000010G                    MOV     #$BFR+G$CUG,R2                                                   ;[TD001]
 431 001006 012700  000030'                    MOV     #CUGPCK,R0                                                       ;[TD001]
 432 001012                                    CALL    MOVE                                                             ;[TD001]
 433                                   ;                                                                                ;[TD001]
 434 001016 016767  000000G 000012G            MOV     DTEDES,G$DTE+$BFR       ; Insert pointer to DTE                  ;[TD001]
 435 001024 016767  000032' 000014G            MOV     CUGFLG,G$FLG+$BFR       ; Insert CUG flags                       ;[TD001]
 436                                   ;                                                                                ;[TD001]
 437 001032 012700  000001'                    MOV     #CUGNAM,R0              ; Address of source CUG name             ;[TD001]
 438 001036 116701  000000'                    MOVB    CUGNML,R1               ; Length of CUG name                     ;[TD001]
 439 001042 010402                             MOV     R4,R2                                                            ;[TD001]
 440 001044 062702  000000G                    ADD     #$BFR,R2                ; Get address to receive CUG name        ;[TD001]
 441 001050                                    CALL    MOVE                                                             ;[TD001]
 442                                   ;                                                                                ;[TD001]
 443 001054 005703                             TST     R3                      ; Test offset to receive length field    ;[TD001]
 444 001056 001403                             BEQ     50$                     ; If zero, no length required            ;[TD001]
 445 001060 116763  000000' 000000G            MOVB    CUGNML,$BFR(R3)         ; Store length field                     ;[TD001]
 446                                   ;                                                                                ;[TD001]
 447 001066                            50$:    PUTAD                           ; Write CUG name block to image          ;[TD001]
 448 001072 016700  000000G                    MOV     $PSIPT,R0               ; Find head of chain pointer             ;[TD001]
 449 001076 062700  000010                     ADD     #H$CUG,R0                                                        ;[TD001]
 450 001102                                    CALL    LNKEND                  ; Insert block in chain                  ;[TD001]
 451 001106 000241                             CLC                             ; Indicate success                       ;[TD001]
 452 001110                            99$:    RESMAP                          ; Restore original mapping state         ;[TD001]
 453 001114                                    RETURN                                                                   ;[TD001]
 454                                                                                                                    ;[TD001]
```

```
    124                                         .SBTTL  TPARS STATE TABLES
    125                                 ;
    126                                 ; TPARS STATE TABLES
    127                                 ;
    128 000060                                  ISTAT$  DSAST,DSAKW
    129                                 ;
    130                                 ; DSA$DF
    131                                 ;
    132 000060                                  STATE$  DSADF
    133 000060                                   TRAN$   %DSA$DF%,,,1,SYNERR
    134
    135 000060                                  STATE$
    136 000060                                   TRAN$  !END,$EXIT
    137 000060                                   TRAN$   <',>,SLO
    138 000060                                   TRAN$   $DIGIT,DARDNX,DARDST           ; START OF REMOTE DTE ADDRESS
    139 000060                                   TRAN$  <'*>,,DARDST
    140
    141 000060                                  STATE$  DARDNX
    142 000060                                   TRAN$   $DIGIT,DARDNX,DARDNO
    143 000060                                   TRAN$  <'*>,DARDNX,DARDNO
    144 000060                                   TRAN$  !END,$EXIT,DARDND
    145 000060                                   TRAN$   <',>,,DARDND           ; END OF REMOTE DTE ADDRESS
    146
    147 000060                                  STATE$  SLO
    148 000060                                   TRAN$   <',>,SHI
    149 000060                                   TRAN$   $NUMBR,,DASSLO ; LOW END OF SUBADDRESS RANGE
    150
    151 000060                                  STATE$
    152 000060                                   TRAN$   !END,$EXIT
    153 000060                                   TRAN$   <',>
    154
    155 000060                                  STATE$  SHI
    156 000060                                   TRAN$   <',>,DACUG
    157 000060                                   TRAN$   $NUMBR,,DASSHI         ; HIGH END OF SUBADDRESS RANGE
    158
    159 000060                                  STATE$
    160 000060                                   TRAN$   !END,$EXIT
    161 000060                                   TRAN$   <',>
    162
    163 000060                                  STATE$  DACUG
    164 000060                                   TRAN$   $STRNG,END,DACUNM      ; CUG ADDRESS
```

```
        .TITLE  VFPDSN - INITIAL CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/
        .ENABL  LC
;
; COPYRIGHT (C) 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       Parse tables and action routines for processing
;       CETAB DSN$DF macro.
;
; EUROPEAN DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  11-OCT-82
;       First written for X25 Gateway RSX Host Access
;       [Tim Dixon EDSG]
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
VFPDSN - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:34  Page 16-1
Symbol table

A$$CHK= 000000          D$$BUG= 177514          MAXSTR= 000020          Q$$OPT= 000010          X$$DBT= 000000
A$$CPS= 000000          D$$ISK= 000000          MOVE  = ****** GX       RNACCT= ****** GX       $ALPHA= 000022
A$$PRI= 000000          D$$L11= 000001          MXACAC= 000020          RNAL  = ****** GX       $ANY  = 000020
A$$TRP= 000000          D$$YNC= 000000          MXACPA= 000010          RNNL  = ****** GX       $BLANK= 000006
BNQST   000176R     003 D$$YNM= 000000          MXACUS= 000020          RNNODE= ****** GX       $DIGIT= 000024
BRA   = 000074          END     000136R     003 MXNOID= 000006          RNPASS= ****** GX       $DNUMB= 000014
BRAST   000170R     003 E$$XPR= 000000          M$$CRB= 000124          RNPL  = ****** GX       $EOS  = 000012
CERR  = ****** GX       FMT10 = ****** GX       M$$CRX= 000000          RNUL  = ****** GX       $ERRMC  000000R     002
CFGBF = ****** GX       FM.10 = 000000          M$$FCS= 000000          RNUSER= ****** GX       $ERR1T= ****** GX
CFGSZ = ****** GX       F$$LVL= 000001          M$$MGE= 000000          RTSPC = ****** GX       $EXIT = 000000
CFLIN = ****** GX       G$$TPP= 000000          M$$NET= 000000          R$$DER= 000000          $FAIL = 177777
CLRSTR  000264R         G$$TSS= 000000          M$$OVR= 000000          R$$K11= 000001          $LAMDA= 000000
COMENT= 000073          G$$TTK= 000000          NQST    000250R     003 R$$SND= 000000          $NUMBR= 000002
COMMA = 000054          G$$WRD= 000000          N$$ACC= 000001          R$$11M= 000000          $QDSN   000000RG
C$$CKP= 000000          I$$RAR= 000000          N$$BUF= 000001          SKIPSP  000126R     003 $RAD50= 000016
C$$ORE= 000400          I$$RDN= 000000          N$$LDV= 000001          STORCH  000272R         $STRNG= 000004
C$$RSH= 177564          KET   = 000076          N$$MCP= 000001          STRBUF  000041R     002 $SUBXP= 000010
DBLQT   000236R     003 K$$CNT= 177546          N$$MLL= 000001          STRING  000146R     003 $$$FLG= 177777
DQUOTE= 000042          K$$CSR= 177546          N$$MOV= 000010          STRLEN  000040R     002 $$$KEY= 000000
DSNACT  000222R         K$$LDC= 000000          N$$NCT= 000001          SYNERR= ****** GX       $$$STA= 000000
DSNDF   000000R     003 K$$TPS= 000074          N$$PEM= 000001          S$$WRG= 000000          $$$TMP= 000000R     005
DSNKW   000000RG    004 LD$LP = 000000          OPSEP   000110R     003 S$$YSZ= 007600          .PCHAR= ****** GX
DSNNOD  000054R         L$$ASG= 000000          P$$P45= 000000          T$$KMG= 000000          .PSTCN= ****** GX
DSNPAS  000160R         L$$DRV= 000000          P$$WRD= 000000          T$$MIN= 000000          .PSTPT= ****** GX
DSNST   000000RG    003 L$$P11= 000001          QBRAST  000214R     003 V$$CTR= 001000          .TPARS= ****** GX
DSNUSR  000116R         L$$11R= 000000          QST     000224R     003

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000322    001   (RW,I,LCL,REL,CON)
DATA    000062    002   (RW,D,LCL,REL,CON)
$STATE  000270    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12488 Words  ( 49 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:48.19
SY:VFPDSN.V2,[132,134]VFPDSN/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPDSN
```

```

                                        ;
                                        ; PRIORITY
                                        ;
                                                .ENABL  LSB
 000162  012700  000000G                DSPRI:  MOV     #DSTPRI,R0              ; GET ADDRESS OF STORAGE FOR PRIORITY
 000166  000402                                 BR      10$                     ; CONTINUE IN COMMON CODE
                                        ;
                                        ; OBJECT NUMBER
                                        ;
 000170  012700  000000G                DSOBJ:  MOV     #DSTOBJ,R0              ; GET ADDRESS OF OBJECT NUMBER STORAGE
 000174  005767  000000G                10$:    TST     .PNUMH                  ; LEGAL VALUE?
 000200  001006                                 BNE     101$                    ; BR IF NO
 000202  105767  000001G                        TSTB    .PNUMB+1                ; IS IT A BYTE VALUE?
 000206  001003                                 BNE     101$                    ; BR IF NO
 000210  116710  000000G                        MOVB    .PNUMB,(R0)             ; STORE SPECIFIED VALUE
 000214                                         RETURN
                                        ;
                                        ; ERRORS
                                        ;
 000216  022700  000000G                101$:   CMP     #DSTOBJ,R0              ; IS THIS FOR OBJECT NUMBER?
 000222  001403                                 BEQ     102$                    ; BR IF YES
 000224                                         EMSG$R  N5                      ; ELSE ILLEGAL PRIORITY
 000232                                 102$:   EMSG$R  N6                      ; ILLEGAL OBJECT NUMBER
                                                .DSABL  LSB
```

```
                                 .SBTTL  MOVBCD - Move variable-length BCD fields                  ;[TD001]
                         ;+                                                                        ;[TD001]
                         ; *** MCVBCD - Move variable-length BCD fields                            ;[TD001]
                         ;                                                                         ;[TD001]
                         ; FUNCTION                                                                ;[TD001]
                         ;       Given table of pointers to variable length fields                 ;[TD001]
                         ;       (data with one-byte precount of number of digits), move all fields ;[TD001]
                         ;       contiguously to output buffer                                     ;[TD001]
                         ;                                                                         ;[TD001]
                         ; INPUTS                                                                  ;[TD001]
                         ;       R0 - Address of table of pointers to fields                       ;[TD001]
                         ;       R1 - Address of output buffer                                     ;[TD001]
                         ;                                                                         ;[TD001]
                         ; OUTPUTS                                                                 ;[TD001]
                         ;       R1 - Next free byte in output buffer                              ;[TD001]
                         ;       R0, R2, R3 destroyed                                              ;[TD001]
                         ;-                                                                        ;[TD001]
001704                   MOVBCD:                                                                   ;[TD001]
001704  012002           5$:     MOV     (R0)+,R2                ; Get next table entry            ;[TD001]
001706  001410                   BEQ     20$                     ; Zero at end of table            ;[TD001]
001710  112203                   MOVB    (R2)+,R3                ; Get length of entry             ;[TD001]
001712  001774                   BEQ     5$                      ; If empty, look for next         ;[TD001]
001714  006203                   ASR     R3                      ; Convert number of digits to bytes ;[TD001]
001716  005503                   ADC     R3                                                        ;[TD001]
001720  112221           10$:    MOVB    (R2)+,(R1)+             ; Move data to output area        ;[TD001]
001722  005303                   DEC     R3                                                        ;[TD001]
001724  003375                   BGT     10$                                                       ;[TD001]
001726  000766                   BR      5$                      ; loop                            ;[TD001]
                         ;                                                                         ;[TD001]
001730                   20$:    RETURN                                                            ;[TD001]
```

```
    101                                          .SBTTL  LOCAL DATA
    102
    103                                  ;
    104                                  ; LOCAL DATA
    105                                  ;
    106
    107 000000                                   .PSECT  DATA,D
    108
    109                                          .NLIST  BEX
    110
    111                                  ;
    112                                  ; LOCAL DATA FOR DTE$DF MACRO
    113                                  ;
    114         000017                           DTEALN = 15.            ; MAXIMUM NUMBER OF DIGITS IN DTE ADDRESS
    115 000000                           DTEADD: .BLKB   DTEALN          ; TEMPORARY STORAGE FOR UNPACKED DTE ADDRESS
    116         000016'                          DTEND = .-1
    117 000017                           DTEPCK: .BLKB   8.              ; PACKED DTE ADDRESS
    118         000006                           DTNMMX = 6.             ; MAX CHARACTERS IN NETWORK NAME
    119 000027                           DTNTNM: .BLKB   DTNMMX          ; NETWORK NAME
    120                                          .EVEN
    121
```

```
VFPDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:36  Page 18-1
Symbol table

A$$CHK= 000000        G$$WRD= 000000        LF.TIM= 000010        L$NMAC  000076        N$$MLL= 000001
A$$CPS= 000000        HF$DLM= 000002        LF.UNL= 020000        L$NMAS  000074        N$$MDV= 000010
A$$PRI= 000000        HF$GWY= 000010        LF.X2P= 000000        L$NMST  000003        N$$NCT= 000001
A$$TRP= 000000        HF$HDS= 000004        LINCTL= ****** GX     L$NNRE  000106        N$$PEM= 000001
BIASX   001322R       HF$XDF= 000020        LINID   000124R   003 L$NRBY  000046        PCKBCD= ****** GX
CERR  = ****** GX     HSHADD= ****** GX     LINID1  000146R   003 L$NRCA  000066        PF.PSV= ****** GX
CFGBF = ****** GX     HSHMN = 000040        LINNAM= ****** GX     L$NRFS  000100        PF.XDF= ****** GX
CFGSZ = ****** GX     HSHMX = 001000        LINUNT= ****** GX     L$NRPK  000056        P$$P45= 000000
CFLIN = ****** GX     HSHSZ = ****** GX     LNKEND= ****** GX     L$NRRE  000105        P$$WRD= 000000
CHKDEV  001206R       HSHTBL  000256R       LN$OFF= 000001        L$NRST  000107        Q$$DPT= 000010
CHKDTE  001042R       H$CUG   000010        LN$DN = 000000        L$NTBY  000052        RTSPC = ****** GX
CNTTIM  000326R       H$DST   000012        LN$SHU= 000002        L$NTCA  000070        R$R0  = 000002
COUNT = ****** GX     H$D29   000014        LN.CLO= 000000        L$NTFS  000102        R$R1  = 000004
CTIM  = ****** GX     H$FLG   000000        LN.DUM= 000005        L$NTPK  000062        R$R2  = 000006
C$$CKP= 000000        H$GLEN  000104        LN.LDA= 000004        L$DMST  000002        R$R3  = 000010
C$$DRE= 000400        H$GLT   000044        LN.LDO= 000003        L$PCS   000112        R$R4  = 000012
C$$RSH= 177564        H$GNAM  000050        LN.OAU= 000003        L$PLS   000110        R$R5  = 000014
DEVCTL= ****** GX     H$GNML= 000020        LN.DFF= 000001        L$QCNT  000122        R$$DER= 000000
DEVNAM= ****** GX     H$GPT   000046        LN.ON = 000000        L$QUEH  000124        R$$K11= 000001
DEVUNT= ****** GX     H$HITS  000034        LN.DOP= 000004        L$QUET  000130        R$$SND= 000000
DE.DFF= ****** GX     H$HLEN  000044        LN.DPE= 000001        L$RTRY  000007        R$$11M= 000000
DE.DN = ****** GX     H$LBDA  000070        LN.REF= 000002        L$SLN   000004        SF.ACT= 000200
DTEAD   000012R   003 H$LBDN  000072        LN.SER= 000002        L$ST    000005        SF.ENA= 000100
DTEADD  000000R   002 H$LDTE  000002        LN.STA= 000017        L$TCLZ  000044        SF.LPB= 000004
DTEAEN  000140R       H$LEN   000042        LN.SUB= 000360        L$TIM   000006        SF.MFL= 000040
DTEALN= 000017        H$LOTS  000032        LN.TRI= 000006        L$$ASG= 000000        SF.PAC= 000020
DTEAST  000052R       H$NETW  000024        LP$EIP= 002000        L$$DRV= 000000        SF.REA= 000010
DTEDES= ****** GX     H$NML = 000006        LP$ENB= 004000        L$$P11= 000001        SF.SER= 000001
DTEDF   000000R   003 H$NPT   000022        LP$PCT= 001400        L$$11R= 000000        SF.SVC= 000002
DTEDIG  000114R       H$PTB   000020        LP$TMR= 100000        L.COST  000015        SF.UNL= 000040
DTEEND  000412R       H$PVC   000006        LP$UP = 010000        L.CTL   000012        SPACE = 000040
DTEFLG= ****** GX     H$RDTE  000004        LP$WTD= 020000        L.CVA   177776        SPFILL= ****** GX
DTEKW   000000RG  004 H$RNW   000042        LP$WTS= 040000        L.DDM   000002        STRNXT= ****** GX
DTEND = 000016R   002 H$SVC   000036        LS$GDN= 000004        L.DDS   000004        SYNERR= ****** GX
DTENET  000352R       H$TRB   000016        LS$OFF= 000000        L.DLC   000003        SYSFDB= ****** GX
DTENW   000076R   003 H$XAVL  000100        LS$STR= 000002        L.DLM   000006        S$$WRG= 000000
DTEPCK  000017R   002 H$XBIA  000074        LS$UP = 000003        L.DLS   000010        S$$YSZ= 007600
DTEST   000000RG  003 H$X29C  000040        LS$WT = 000001        L.FLG   000000        S.CDST  000001
DTNMMX= 000006        I$$RAR= 000000        L$ACHN  000012        L.KRBA  000016        S.FLG   000000
DTNTNM  000027R   002 I$$RDN= 000000        L$APVC  000014        L.LEN = 000022        S.LEN   000004
D$$BUG= 177514        K$$CNT= 177546        L$ASVC  000010        L.MPF   000022        S.NMST  000002
D$$ISK= 000000        K$$CSR= 177546        L$AUC   000042        L.NMST  000020        S.DWNR  000003
D$$L11= 000001        K$$LDC= 000000        L$CHLS  000034        L.NSTA  000014        T$$KMG= 000000
D$$YNC= 000000        K$$TPS= 000074        L$CHTB  000030        L.OWNR  000021        T$$MIN= 000000
D$$YNM= 000000        LD$LP = 000000        L$CLEN= 000044        L.UNT   000013        V$$CTR= 001000
END     000114R   003 LF.ACT= 100000        L$CTEN  000032        M$$CRB= 000124        X$$DBT= 000000
E$$XPR= 000000        LF.BRD= 000400        L$CTIM  000040        M$$CRX= 000000        $ALPHA= 000022
FMT10 = ****** GX     LF.BWT= 000007        L$DTEA  000020        M$$FCS= 000000        $ALPDL= ****** GX
FMT8  = ****** GX     LF.ENA= 002000        L$DTEL  000017        M$$MGE= 000000        $ANY  = 000020
FM.10 = 000000        LF.LPB= 001000        L$GLEN  000134        M$$NET= 000000        $BFR  = ****** GX
FM.8  = 000000        LF.MDC= 000100        L$LEN   000122        M$$DVR= 000000        $BLANK= 000006
FNDPDV= ****** GX     LF.MFL= 004000        L$LLCH  000016        NEXT  = ****** GX     $CEACC= ****** GX
F$$LVL= 000001        LF.MTP= 000020        L$LNK   000000        NUMDTE= 000020        $DIGIT= 000024
F.RSIZ= ****** GX     LF.PAC= 000200        L$MCHN  000036        N$$ACC= 000001        $DNUMB= 000014
G$$TPP= 000000        LF.RDY= 040000        L$NETW  000114        N$$BUF= 000001        $EOS  = 000012
G$$TSS= 000000        LF.REA= 010000        L$NIRE  000072        N$$LDV= 000001        $ERROD  000154R   002
G$$TTK= 000000        LF.SER= 000040        L$NLRE  000104        N$$MCP= 000001        $ERROP  000214R   002
```

```
    103                                          .SBTTL  ERROR MESSAGES
    104
    105 000030                                   NTLER$  ,P1,10,CERR,RTSPC,CFLIN,<occurrence of PSN$DF>                  ;**-2
    106 000066                                   NTLER$  ,P2,10,CERR,RTSPC,CFLIN,<network name>
    107 000114                                   NTLER$  ,P3,10,CERR,RTSPC,CFLIN,<number of ports>
    108 000144                                   NTLER$  ,P4,10,CERR,RTSPC,CFLIN,<flags word>
    109 000170                                   NTLER$  ,P5,8,CERR,RTSPC,CFLIN,<PSI home block allocation failure>
    110 000242                                   NTLER$  ,P6,8,CERR,RTSPC,CFLIN,<Port table allocation failure>
    111 000310                                   NTLER$  ,PA,8,CERR,RTSPC,CFLIN,<PSV mapping tables allocation failure>  ;[TD001]
    112 000366                                   NTLER$  ,D1,10,CERR,RTSPC,CFLIN,<low subaddresss value>
    113 000424                                   NTLER$  ,D2,10,CERR,RTSPC,CFLIN,<high subaddress value>
    114
    115                                          .EVEN
    116                                          .LIST   BEX                                                             ;**-1
    117 000000                                   .PSECT
```

```
VFPPSN     CREATED BY  MACRO  ON 29-JUN-85 AT 02:37       PAGE 3
SYMBOL CROSS REFERENCE                                    CREF   04.00
SYMBOL   VALUE               REFERENCES

.PNUMH   = ******   GX        11-228      11-239      12-287
.PSTCN   = ******   GX        11-213      11-215
.PSTPT   = ******   GX        11-218
.TPARS   = ******   GX        9-139
```

```
                                                .SBTTL  PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
                                        ;+
                                        ;
                                        ; PCKBCD - PACK A STRING OF DIGITS IN BCD FORMAT
                                        ;
                                        ; INPUTS:
                                        ;       R0 - NUMBER OF BYTES IN PACKED ADDRESS
                                        ;       R4 - ADDRESS OF BUFFER CONTAINING ASCI DIGITS TO PACK (0=END OF BUFFER)
                                        ;       R5 - ADDRESS OF BUFFER TO STORE PACKED DATA
                                        ;       COUNT - NUMBER OF DIGITS SPECIFIED
                                        ;
                                        ; OUTPUTS:
                                        ;       BUFFER SPECIFIED IN R5 CONTAINS DATA IN BCD FORMAT
                                        ;
                                        ;-

001566  010501                          PCKBCD: MOV     R5,R1                   ; GET ADDRESS OF PACKED DATA
001570  116702  000024'                         MOVB    COUNT,R2                ; GET NUMBER OF DIGITS IN STRING
001574  001423                                  BEQ     40$                     ; BR IF NOTHING TO DO
001576  105021                          5$:     CLRB    (R1)+                   ; INITIALIZE PACKED ADDRESS
001600  005300                                  DEC     R0                      ; MORE TO INITIALIZE?
001602  003375                                  BGT     5$                      ; BR IF YES
001604  005001                                  CLR     R1                      ; POSITION INIDCATOR (0=HIGH FOUR BITS)
001606  005305                                  DEC     R5                      ; UPDATE ADDRESS FOR FIRST TIME
001610  112400                          10$:    MOVB    (R4)+,R0                ; GET DIGIT TO PACK
001612  005701                                  TST     R1                      ; STORE IN HIGH FOUR BITS?
001614  001007                                  BNE     20$                     ; BR IF NO
001616                                          ASL$    4,R0                    ; MOVE TO HIGH FOUR BITS
001626  005201                                  INC     R1                      ; INDICATE STORE IN LOW FOUR BITS NEXT
001630  005205                                  INC     R5                      ; POINT TO NEXT BYTE
001632  000401                                  BR      30$                     ; BR TO STORE DECIMAL
001634  005301                          20$:    DEC     R1                      ; INDICATE STORE IN HIGH FOUR BITS NEXT
001636  150015                          30$:    BISB    R0,(R5)                 ; PACK THE NUMBER INTO THE BUFFER
001640  005302                                  DEC     R2                      ; MORE TO CONVERT?
001642  003362                                  BGT     10$                     ; BR IF YES
001644                                  40$:    RETURN
```

```
     55                                              .SBTTL  MACRO DEFINITIONS
     56
     57                                      ;
     58                                      ; LIBRARY MACROS
     59                                      ;
     60                                              .MCALL  ISTAT$,STATE$,TRAN$,EMSG$R,NTLER$,PDVDF$,SLTDF$
     61
     62 000000                                       PDVDF$                          ; DEFINE PDV OFFSETS AND SYMBOLS
     63 000000                                       SLTDF$                          ; DEFINE SLT OFFSETS AND SYMBOLSS
     64                                      ;
     65                                      ; LOCAL MACRO DEFINITIONS
     66                                      ;
     67                                              .MACRO  REJ$
     68                                              ADD     #2,(SP)                 ; RETURN TO CALLER+2
     69                                              CLR     SYNERR                  ; INDICATE NO SYNTAX ERROR
     70                                              .ENDM
```

```
                                                                  D  3
VFGX2P      CREATED BY  MACRO  ON 29-JUN-85 AT 02:32      PAGE 3
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

ZF.MUX   = 000040           #5-62
ZF.PSE   = 002000           #5-62
ZF.SLI   = 010000           #5-62
ZF.TIM   = 000200           #5-62
ZF.X3P   = 000000           #5-62
ZS.ASN   = 100000           #5-62
ZS.BSY   = 140000           #5-62
Z.AVL      000014           #5-62
Z.DAT      000016           #5-62
Z.DSP      000000           #5-62       5-62
Z.FLG      000010           #5-62
Z.LEN    = 000016           #5-62
Z.LLN      000006           #5-62
Z.MAP      000020           #5-62
Z.NAM      000004           #5-62       11-229
Z.PCB      000012           #5-62
Z.SCH      000007           #5-62
$ALPHA   = 000022           #10-136
$ANY     = 000020           #10-136
$BLANK   = 000006           #10-136
$CAT5    = ******   GX        11-204
$DIGIT   = 000024           #10-136
$DNUMB   = 000014           #10-136
$EOS     = 000012           #10-136
$ERR1T   = ******   GX        9-130
$ERR2C   = ******   GX        11-261
$ERR2F   = ******   GX        11-252
$ERR2I   = ******   GX        11-238
$ERR2J     000004 R         #8-98       11-271
$ERR2K     000046 R         #3-99       11-283
$ERR2L     000106 R         #8-100      11-295
$EXIT    = 000000           #10-136
$FAIL    = 177777           #10-136
$GPRM    = ******             10-136
$LAMDA   = 000000           #10-136
$NUMBR   = 000002           #10-136
$PDVTA   = ******   GX        11-227
$QX2P      000144 RG        #9-116
$RAD50   = 000016           #10-136
$RONLY   = ******             10-136    10-136      10-136
$SLTA    = ******   GX        11-225
$STRNG   = 000004           #10-136
$SUBXP   = 000010           #10-136
$X2HLD   = ******   GX      *11-293
$X2MBS   = ******   GX      *11-259
$X2MRT   = ******   GX      *11-269
$X2MWS   = ******   GX      *11-250
$X2RET   = ******   GX      *11-281
$$$FLG   = 177777           #10-136
$$$KEY   = 177777           #10-136
.PNUMB   = ******   GX        11-212    11-214      11-222      11-224      11-245      11-258      11-267      11-269      11-277
                              11-289
```

E 3

```
000270                                     RETURN
000272  005267  000000G            101$:   INC     SYNERR                  ; INDICATE SYNTAX ERROR
000276  062716  000002                     ADD     #2,(SP)                 ; REJECT THE TRANSITION
000302                                     RETURN
000304                             102$:   EMSG$R  2H                      ; ILLEGAL TIMER VALUE
                                   ;
                                   ; END OF TIMER VALUES
                                   ;
000312  126727  000010' 000007     X3.END: CMPB    TIMRCT,#X3TMCT          ; WERE ALL VALUES PROCESSED?
000320  001402                             BEQ     10$                     ; BR IF YES
000322  062716  000002                     ADD     #2,(SP)                 ; REJECT THE TRANSITION
000326                             10$:    RETURN
        000001                             .END
```

```
    197                                                .SBTTL  $FNNAM - FIND REMOTE BY NAME
    198                                        ;+
    199                                        ; *** - $FNNAM - FIND REMOTE NODE BLOCK BY NAME
    200                                        ;
    201                                        ; INPUTS:
    202                                        ;       R1 = ADDRESS OF REMOTE NODE STRING
    203                                        ;
    204                                        ; OUTPUTS:
    205                                        ;       C-BIT = SUCCESS/FAILURE
    206                                        ;       R0 = UNMAPPED ADDRESS OF OBJECT BLOCK, IF FOUND
    207                                        ;       R1 = UNMAPPED ADDRESS OF PREVIOUS
    208                                        ;
    209                                        ;-
    210 000470                                 $FNNAM::
    211 000470  010246                                 MOV     R2,-(SP)                ; SAVE R2
    212
    213 000472                                         SWTCHI  #$HMBUF                 ; USE TEMPORARY BUFFER
    214 000500                                         GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
    215 000520                                         SWTCHI                          ; RESET DEFAULT BUFFER
    216
    217 000526  012700  000004G                        MOV     #$HMBUF+D$RNN+2,R0      ; GET START OF REMOTES LIST
    218
    219 000532  011067  000000G                        MOV     (R0),$BFR               ; GET FIRST IN LIST
    220 000536  010146                                 MOV     R1,-(SP)                ; SAVE ADDRESS OF NAME ON STACK
    221 000540  010067  177234                         MOV     R0,CURUNM               ; SAVE UNMAPPED ADDR OF CURRENT BLOCK
    222 000544  016701  177230                 10$:    MOV     CURUNM,R1               ; SAVE ADDRESS OF PREVIOUS
    223 000550  016767  000000G 177222                 MOV     $BFR,CURUNM             ; GET NEXT OBJECT IN LIST
    224 000556  000261                                 SEC                             ; ASSUME FAILURE
    225 000560  001431                                 BEQ     30$                     ; IF EQ, NO MORE
    226 000562                                         CEACC$  CURUNM,R0               ; CONVERT TO MAPPED ADDRESS
    227 000574                                 20$:    GETAD   R0,#R.LEN               ; READ REMOTE IN INTERNAL BUFFER
    228 000612  011602                                 MOV     (SP),R2                 ; GET ADDRESS OF NODE NAME
    229 000614  022267  000002G                        CMP     (R2)+,$BFR+R.NAM        ; IS THIS THE RIGHT REMOTE ?
    230 000620  001351                                 BNE     10$                     ; IF NE, NO
    231 000622  022267  000004G                        CMP     (R2)+,$BFR+R.NAM+2      ; ARE WE SURE ?
    232 000626  001346                                 BNE     10$                     ; IF NE, NO
    233 000630  022267  000006G                        CMP     (R2)+,$BFR+R.NAM+4      ; ARE WE REALLY SURE ?
    234 000634  001343                                 BNE     10$                     ; IF NE, NO
    235 000636  016700  177136                         MOV     CURUNM,R0               ; GET UNMAPPED ADDRESS OF BLOCK
    236 000642  000241                                 CLC                             ; ELSE, INDICATE SUCCESS
    237 000644  005226                         30$:    INC     (SP)+                   ; CLEAN UP THE STACK
    238 000646  012602                                 MOV     (SP)+,R2                ; RESTORE R2
    239 000650                                         RETURN
    240
    241
    242         000001                                 .END
```

```
     80                                          .SBTTL  LOCAL SYMBOL DEFINITIONS
     81
     82                                   ;
     83                                   ; LOCAL SYMBOL DEFINITIONS
     84                                   ;
     85          000040                          SPACE = 40                ; ASCII SPACE
     86          000006                          CUNMMX = 6.               ; Maximum length of old-format CUG name         ;[TD001]
     87          000020                          MAXID = 16.               ; Maximum length of net. mgmt. id-string        ;[TD001]
```

```
    455         000001                          .END                                                                    ;**-32
```

```
                                .SBTTL  DSC$DF STATE TABLES

                        ;
                        ; DSC$DF
                        ;
000060                          STATE$  DSCDF
000060                           TRAN$   %DSC$DF%,,,1,SYNERR

000060                          STATE$
000060                           TRAN$   !END,$EXIT
000060                           TRAN$   $STRNG,,DCMSK          ; CALL MASK

000060                          STATE$
000060                           TRAN$   <',>

000060                          STATE$  CALVAL
000060                           TRAN$   $STRNG,END,DCVAL       ; CALL VALUE

000060                          STATE$  END
000060                           TRAN$   $EOS,$EXIT
000060                           TRAN$   <';>,$EXIT

000060                          STATE$
```

```
     50                                              .SBTTL  Local Symbols
     51
     52          000042                              DQUOTE = <'">                    ; String delimiter
     53          000054                              COMMA = <',>                     ; Operand separator
     54          000073                              COMENT = <';>                    ; Start-of-comment character
     55          000074                              BRA = 74                         ; Left angle bracket (<)
     56          000076                              KET = 76                         ; Right angle bracket (>)
     57
     58          000006                              MXNOID = 6                       ; Max length of node id
     59          000020                              MXACUS = 16.                     ; Max length of access cntrl user
     60          000010                              MXACPA = 8.                      ; Max length of access cntrl password
     61          000020                              MXACAC = 16.                     ; Max length of access cntrl account
     62
     63          000020                              MAXSTR = 16.                     ; Max length of string seen in parse
```

```
VFPDSN      CREATED BY  MACRO  ON 29-JUN-85 AT 02:35      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

BRA      = 000074           #5-55
CERR     = ******   GX       6-76
CFGBF    = ******   GX       8-108
CFGSZ    = ******   GX       8-107
CFLIN    = ******   GX       6-76
CLRSTR     000264 R         #15-319
COMENT   = 000073           #5-54
COMMA    = 000054           #5-53
DQUOTE   = 000042           #5-52
DSNACT     000222 R         #14-297
DSNDF      000000 R          8-104
DSNKW      000000 RG         8-106      #9-122
DSNNOD     000054 R         #11-226
DSNPAS     000160 R         #13-273
DSNST      000000 RG        #9-122
DSNUSR     000116 R         #12-250
FMT10    = ******   GX       6-76
FM.10    = 000000           #6-76
KET      = 000076           #5-56
MAXSTR   = 000020           #5-63       7-84       16-333
MOVE     = ******   GX       11-233     12-257     13-280     14-304
MXACAC   = 000020           #5-61       14-299
MXACPA   = 000010           #5-60       13-275
MXACUS   = 000020           #5-59       12-252
MXNOID   = 000006           #5-58       11-228
RNACCT   = ******   GX       14-303
RNAL     = ******   GX      *14-301
RNNL     = ******   GX      *11-230
RNNODE   = ******   GX       11-232
RNPASS   = ******   GX       13-279
RNPL     = ******   GX      *13-277
RNUL     = ******   GX      *12-254
RNUSER   = ******   GX       12-256
RTSPC    = ******   GX       6-76
STORCH     000272 R         #16-333
STRBUF     000041 R         #7-84       12-255     13-278     14-302    *16-336
STRLEN     000040 R         #7-83       12-251     13-274     14-298    *15-319     16-333     16-335    *16-337
SYNERR   = ******   GX      *8-109     *8-112
S$$BAS   = ******            6-76       6-76
$ALPHA   = 000022           #9-122
$ANY     = 000020           #9-122
$BLANK   = 000006           #9-122
$DBG$    = ******           #   03
$DIGIT   = 000024           #   122
$DNUMB   = 000014           # -122
$EOS     = 000012           #9-122
$ERRMC     000000 R         #6-76       11-236     12-260     13-283     14-307
$ERR1T     ******   GX       8-118
$EXIT      000000           #9-122
$FAIL      177777           #9-122
$GPRM      ******            9-122
$LAMDA     000000           #9-122
```

```

                                     ;
                                     ; TASK NAME
                                     ;
 000240  012700  000000G             DSTSK:  MOV     #DSTTKL,R0              ; address of task name buffer            ;[TD001]
 000244  012701  000006                      MOV     #DSTKMX,R1              ; max length of old-format task name     ;[TD001]
 000250                                      CALL    STORID                  ; check and store task name              ;[TD001]
 000254  103401                              BCS     101$                    ; branch if bad name                     ;[TD001]
 000256                                      RETURN                                                                   ;**-11
                                     ;
                                     ; ERRORS
                                     ;
 000260                              101$:   EMSG$R  N7                      ; ILLEGAL TASK NAME
```

```
                                .SBTTL  MOVLEN - Move length fields                                      ;[TD001]
                        ;+                                                                               ;[TD001]
                        ; *** MOVLEN - Move length fields                                                ;[TD001]
                        ;                                                                                ;[TD001]
                        ; FUNCTION                                                                       ;[TD001]
                        ;       Given table of pointers to counts (one byte) of                          ;[TD001]
                        ;       variable fields, move all counts contiguously to                         ;[TD001]
                        ;       output buffer.                                                           ;[TD001]
                        ;                                                                                ;[TD001]
                        ; INPUTS                                                                         ;[TD001]
                        ;       R0 - Address of table of pointers to counts                              ;[TD001]
                        ;       R1 - Address of output buffer                                            ;[TD001]
                        ;                                                                                ;[TD001]
                        ; OUTPUTS                                                                        ;[TD001]
                        ;       R1 - Address of next free byte in output buffer                          ;[TD001]
                        ;       R0, R2 Destroyed                                                         ;[TD001]
                        ;-                                                                               ;[TD001]
001732                  MOVLEN:                                                                          ;[TD001]
001732  012002          5$:     MOV     (R0)+,R2                ; Get table entry                        ;[TD001]
001734  001402                  BEQ     20$                     ; Zero at end of table                   ;[TD001]
001736  111221                  MOVB    (R2),(R1)+              ; Move length field                      ;[TD001]
001740  000774                  BR      5$                      ; loop                                   ;[TD001]
                        ;                                                                                ;[TD001]
001742                  20$:    RETURN                                                                   ;[TD001]

        000001                  .END
```

```
    123                                        .SBTTL  ERROR MESSAGES
    124
    125 000036                                 NTLER$  ,03,10,CERR,RTSPC,CFLIN,<DTE address value>                          ;**-1
    126 000070                                 NTLER$  ,04,10,CERR,RTSPC,CFLIN,<hash table size>
    127 000120                                 NTLER$  ,07,8,CERR,RTSPC,CFLIN,<Line not in system>
    128 000154                                 NTLER$  ,00,10,CERR,RTSPC,CFLIN,<network name parameter>
    129 000214                                 NTLER$  ,0P,8,CERR,RTSPC,CFLIN,<DTE descriptor allocation failure>
    130 000266                                 NTLER$  ,0Q,8,CERR,RTSPC,CFLIN,<Too many DTE$DFs>
    131                                        .LIST   BEX                                                                   ;**-1
    132                                        .EVEN
    133 000000                                 .PSECT
```

```
VFPDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:36  Page 18-2
Symbol table

$ERROQ  000266R     002 $GETAD= ****** GX        $PUTAD= ****** GX        $SLTMA= ****** GX        $$$STA= 000000
$ERRO3  000036R     002 $GETRV= ****** GX        $QDTE   000000RG         $SLTNM= ****** GX        $$$TMP= 000012R      005
$ERRO4  000070R     002 $LAMDA= 000000           $RADDR= ****** GX        $STRNG= 000004           .PNUMB= ****** GX
$ERRO5= ****** GX       $LLCTA= ****** GX        $RAD50= 000016           $SUBXP= 000010           .PNUMH= ****** GX
$ERRO7  000120R     002 $NUMBR= 000002           $RBLCK= ****** GX        $$$FLG= 177777           .PSTCN= ****** GX
$ERR1T= ****** GX       $PFLAG= ****** GX        $SAVAL= ****** GX        $$$KEY= 000002           .PSTPT= ****** GX
$EXIT = 000000          $PSIPT= ****** GX        $SAVRG= ****** GX        $$$R  = 000010           .TPARS= ****** GX
$FAIL = 177777

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001342    001   (RW,I,LCL,REL,CON)
DATA    000320    002   (RW,D,LCL,REL,CON)
$STATE  000154    003   (RW,D,LCL,REL,CON)
$KTAB   000006    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 1
Work  file writes: 2
Size of work file: 16924 Words  ( 67 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:47.32
SY:VFPDTE.V2,[132,134]VFPDTE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPDTE
```

```
    119                                         .SBTTL  LOOK FOR PSN$DF MACRO
    120
    121                                 ;+
    122                                 ; $QPSN - LOOK FOR PSN$DF MACRO
    123                                 ;
    124                                 ; INPUTS:
    125                                 ;       NONE
    126                                 ;
    127                                 ; OUTPUTS:
    128                                 ;       C-BIT=SUCCESS/FAILURE
    129                                 ;       R3,R4,R5=DESTROYED
    130                                 ;-
    131                                                                                                         ;[TD001]
    132 000000                          $QPSN::                                                                 ;[TD001]
    133 000000  012705  000000'                 MOV     #PSNDF,R5       ; STATE TABLE ADDRESS                   ;[TD001]
    134 000004  005001                          CLR     R1              ; FULL KEYWORD MATCH LENGTH             ;**-1
    135 000006  012702  000000'                 MOV     #PSNKW,R2       ; KEYWORD TABLE ADDRESS
    136 000012  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
    137 000016  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
    138 000022  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
    139 000026                                  CALL    .TPARS          ; GO DO THE PARSE
    140 000032  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
    141 000034  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
    142 000040  001401                          BEQ     101$            ; IF EQ, YES
    143 000042                          20$:    RETURN
    144
    145                                 ;
    146                                 ; ERROR CONDITION
    147                                 ;
    148 000044                          101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
VFPPSN     CREATED BY  MACRO  ON 29-JUN-85 AT 02:37      PAGE 4

MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME         REFERENCES

CALL                 9-139     11-212     11-219     12-273     13-302     13-311     13-320     13-343     13-348     13-355
                    14-386     14-390     14-398     14-403     15-427     15-431     15-432     15-436     15-437     15-439
                    15-442
CEACC$             #5-62       14-390     15-431     15-436
DBGTP$             #10-154    #10-173    #10-177    #10-181    #10-185    #10-189    #10-196
EMSG$R             #5-61        9-148     11-262     11-263     11-264     11-265     12-282     13-362     13-363     13-364
                    13-365
GETAD              #5-62
GETRV              #5-61       15-439
ISTAT$             #5-62       10-154
MTRAN$             #10-154
NTLER$             #5-61        8-105      8-106      8-107      8-108      8-109      8-110      8-111      8-112      8-113
PHBDF$             #5-61        5-68
PUTAD              #5-62       14-398     15-432     15-437
PUTRC              #5-61       13-343     14-403     15-442
RAND               #13-343     13-343    #14-398     14-398    #14-403     14-403    #15-432     15-432    #15-437     15-437
                   #15-439     15-439    #15-442     15-442
RESMAP             #5-77       14-399     15-438
RESRG              #5-61       13-342
RETURN               9-143     11-223     11-235     11-258     12-278     12-296     13-358     14-405     15-444
SAVMAP             #5-73       14-389     15-430
SAVRG              #5-61       13-322
STATE$             #5-62       10-159    #10-162    #10-165    #10-165    #10-171    #10-175    #10-179    #10-183    #10-187
                   #10-191    #10-194    #10-198
TRAN$              #5-62      #10-160    #10-163    #10-166    #10-169    #10-172    #10-173    #10-176    #10-177    #10-180
                   #10-181    #10-184    #10-185    #10-188    #10-189    #10-192    #10-195    #10-196
VNPDBG             #5-63       11-206
```

```
                                        .SBTTL  STRNXT - STORE NEXT DIGIT

                                ;+
                                ;
                                ; STRNXT - STORE NEXT DIGIT
                                ;
                                ; INPUTS:
                                ;       .PCHAR - MATCHED CHARACTER FROM TPARS TRANSITION
                                ;       NEXT - ADDRESS OF NEXT BYTE AVAILABLE FOR STORAGE
                                ;
                                ; OUTPUTS:
                                ;       NEXT - ADDRESS OF NEXT AVAILABLE BYTE
                                ;       COUNT - UPDATED
                                ;       R0,R1 DESTROYED
                                ;-
001646  116767  000000G 000054' STRNXT: MOVB    .PCHAR,CVTBUF           ; MOVE DIGIT TO CONVERTSION BUFFER
001654  012700  000054'                 MOV     #CVTBUF,R0              ; POINT TO CONVERSION BUFFER
001660                                  CALL    $CDTB                   ; CONVERT TO BINARY
001664  110177  000000'                 MOVB    R1,@NEXT                ; STORE BINARY DIGIT
001670  005267  000000'                 INC     NEXT                    ; POINT TO NEXT AVAILABLE BYTE
001674  105267  000024'                 INCB    COUNT                   ; INCREMENT DIGIT COUNT
001700                                  RETURN

        000001                          .END
```

```
     72                                          .SBTTL  CONSTANT DEFINITIONS
     73
     74                                  ;
     75                                  ; CONSTANT DEFINITIONS
     76                                  ;
     77        000007                            X2WNMX = 7                ; MAXIMUM WINDOW SIZE FOR X2P$DF
     78        000001                            X2WNMN = 1                ; MINIMUM WINDOW SIZE FOR X2P$DF
     79        001130                            X2RTMX = 600.             ; MAXIMUM VALUE FOR RETRANSMIT VALUE
     80        001130                            X2HLMX = 600.             ; MAXIMUM VALUE FOR HOLDBACK TIMER
```

```
                                                                   E  3
VFGX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32      PAGE 4
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE            REFERENCES

.PNUMH  = ******  GX      11-210     11-220     11-243     11-256     11-265     11-275     11-287
.PSTCN  = ******  GX      11-201
.PSTPT  = ******  GX      11-203
.TPARS  = ******  GX      9-122
```

F 3

```
                                                                E 4
VFGX3P - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 02:32  Page 11-2
Symbol table

A$$CHK= 000000     F$$LVL= 000001     N$$LDV= 000001         X3DEV   000002R    002 $ERR2E  000122R    002
A$$CPS= 000000     G$$TPP= 000000     N$$MCP= 000001         X3NXT   000042R    003 $ERR2F= ****** GX
A$$PRI= 000000     G$$TSS= 000000     N$$MLL= 000001         X3PDF   000000R    003 $ERR2G  000156R    002
A$$TRP= 000000     G$$TTK= 000000     N$$MOV= 000010         X3PKW   000000RG   004 $ERR2H  000232R    002
BLKSMN= 000040     G$$WRD= 000000     N$$NCT= 000001         X3PST   000000RG   003 $EXIT = 000000
BLKSMX= 002000     I$$RAR= 000000     N$$PEM= 000001         X3TMCT= 000007         $FAIL = 177777
CERR  = ****** GX  I$$RDN= 000000     P$$P45= 000000         X3UNT   000006R    002 $LAMDA= 000000
CFGBF = ****** GX  K$$CNT= 177546     P$$WRD= 000000         X3.BEG  000212R        $NUMBR= 000002
CFGSZ = ****** GX  K$$CSR= 177546     Q$$OPT= 000010         X3.DBS  000052R        $QX3P   000000RG
CFLIN = ****** GX  K$$LDC= 000000     RTSPC = ****** GX      X3.DWS  000132R        $RAD50= 000016
CHKBLK= ****** GX  K$$TPS= 000074     R$$DER= 000000         X3.END  000312R        $STRNG= 000004
CHKWND= ****** GX  LD$LP = 000000     R$$K11= 000001         X3.MBS  000074R        $SUBXP= 000010
C$$CKP= 000000     L$$ASG= 000000     R$$SND= 000000         X3.MWS  000154R        $X3CAL= ****** GX
C$$ORE= 000400     L$$DRV= 000000     R$$11M= 000000         X3.NXT  000224R        $X3DBS= ****** GX
C$$RSH= 177564     L$$P11= 000001     SYNERR= ****** GX      $ALPHA= 000022         $X3DWS= ****** GX
D$$BUG= 177514     L$$11R= 000000     S$$WRG= 000000         $ANY  = 000020         $X3MBS= ****** GX
D$$ISK= 000000     M$$CRB= 000124     S$$YSZ= 007600         $BLANK= 000006         $X3MWS= ****** GX
D$$L11= 000001     M$$CRX= 000000     TIMRCT  000010R    002 $DIGIT= 000024         $$$FLG= 177777
D$$YNC= 000000     M$$FCS= 000000     T$$KMG= 000000         $DNUMB= 000014         $$$KEY= 000000
D$$YNM= 000000     M$$MGE= 000000     T$$MIN= 000000         $EOS  = 000012         $$$STA= 000042R    003
E$$XPR= 000000     M$$NET= 000000     V$$CTR= 001000         $ERR1T= ****** GX      $$$TMP= 000000R    005
FMT8  = ****** GX  M$$OVR= 000000     WNDSMX= 000177         $ERR2B  000012R    002 .PNUMB= ****** GX
FMT8B = ****** GX  NEXT    000000R    002 X$$DBT= 000000     $ERR2C= ****** GX      .PNUMH= ****** GX
FM.8  = 000000     N$$ACC= 000001     X3CTL   000004R    002 $ERR2D  000046R    002 .TPARS= ****** GX
FM.8B = 000000     N$$BUF= 000001

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000330    001   (RW,I,LCL,REL,CON)
DATA    000256    002   (RW,D,LCL,REL,CON)
$STATE  000060    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12456 Words  ( 49 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:26.12
SY:VFGX3P.V2,[132,134]VFGX3P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFGX3P
```

F 4

```
Symbol table

A$$CHK= 000000            F.RSIZ= ****** GX         LO.SKA  000014            L$$ASG= 000000            OP$MCO= 000002
A$$CPS= 000000            F.URBD= ****** GX         LO.SKN  000014            L$$DRV= 000000            OP$MDE= 000400
A$$PRI= 000000            G$$TPP= 000000            LO.SNM  000035            L$$P11= 000001            OP$MLI= 000004
A$$TRP= 000000            G$$TSS= 000000            LO.TPA  000020            L$$11R= 000000            OP$MON= 000100
A.ACC   000020            G$$TTK= 000000            LO.TRB  000034            M$$CRB= 000124            OP$MST= 000020
A.LNK   000000            G$$WRD= 000000            LO.TRI  000030            M$$CRX= 000000            OP$NAM= 000004
A.NAM   000002            I$$RAR= 000000            LO.UNT  000037            M$$FCS= 000000            OP$NET= 004000
A.PUD   000010            I$$RDN= 000000            LO.VCT  000014            M$$MGE= 000000            OP$NLI= 000002
A.REM   000012            K$$CNT= 177546            LO.VER  000042            M$$NET= 000000            OP$NOD= 000020
A.UCB   000010            K$$CSR= 177546            LR.ACC  000020            M$$OVR= 000000            OP$OUT= 002000
CURUNM  000000R           K$$LDC= 000000            LR.ACO  000052            N$$ACC= 000001            OP$OWN= 000020
C$$CKP= 000000            K$$TPS= 000074            LR.ADD  000012            N$$BUF= 000001            OP$PAR= 000010
C$$ORE= 000400            LD$LP = 000000            LR.ALI  000002            N$$LDV= 000001            OP$PRI= 000010
C$$RSH= 177564            LD.ALI= 000030            LR.BLK  000010            N$$MCP= 000001            OP$RET= 010000
D$AMXC  000072            LD.CEX= 000024            LR.CTL  000004            N$$MLL= 000001            OP$RPA= 000010
D$AMXH  000074            LD.CIR= 000035            LR.DES  000010            N$$MOV= 000010            OP$SCO= 000001
D$ANN   000000            LD.LIN= 000025            LR.HLP  000002            N$$NCT= 000001            OP$SEG= 000400
D$BRPR  000102            LD.LOG= 000031            LR.LIN  000002            N$$PEM= 000001            OP$SER= 000400
D$BRTM  000100            LD.MOD= 000037            LR.NAM  000012            OF$INS= 000001            OP$SIN= 000010
D$DELF  000045            LD.NOD= 000027            LR.PAS  000041            OF$LOG= 000100            OP$STA= 000004
D$DELW  000046            LD.OBJ= 000032            LR.PRO  000002            OF$OFF= 000002            OP$TPA= 000020
D$END = 000104            LD.PRO= 000026            LR.STS  000000            OF$ON = 000000            OP$TRI= 000040
D$FNB   000034            LO.ADD  000022            LR.TRI  000006            OF$SMC= 000200            OP$UCS= 000020
D$HIOR  000024            LO.ADR  000052            LR.TYP  000002            OF.PRO= 000040            OP$USE= 000002
D$HOST  000022            LO.CLS  000002            LR.UCB  000016            OF.RLU= 000100            OP$VEC= 000040
D$INAC  000044            LO.CON  000006            LR.UIC  000020            OF.SMC= 000200            OP$VER= 000040
D$INCT  000042            LO.COP  000010            LR.UNT  000005            OP$ADD= 000200            OP$XAC= 000001
D$IPL   000051            LO.COS  000020            LS.ALI= 000016            OP$ALL= 000200            OP$XPR= 000002
D$LID   000020            LO.CSR  000010            LS.CEX= 000001            OP$CCS= 000004            OP$X5S= 000004
D$LNAM  000006            LO.CTL  000032            LS.CIR= 000033            OP$CIE= 004000            OP$X9S= 000010
D$LNUM  000014            LO.DES  000002            LS.CXO= 010000            OP$CON= 000400            O.FLG   000003
D$LST   000047            LO.DTE  000047            LS.CX5= 000400            OP$COP= 000001            O.LEN   000012
D$MAXC  000064            LO.EVT  000004            LS.ECH= 001000            OP$COS= 000100            O.LNK   000000
D$MAXH  000066            LO.GRO  000024            LS.FDX= 004000            OP$DST= 000200            O.MXC   000004
D$MAXV  000070            LO.HOS  000040            LS.HDX= 010000            OP$DTE= 000040            O.NAM   000006
D$MLL   000040            LO.HTM  000032            LS.HLP= 000012            OP$DUP= 000002            O.TYP   000002
D$MNOD  000041            LO.INA  000062            LS.LIN= 000003            OP$EST= 000100            P$$P45= 000000
D$NA    000062            LO.INC  000060            LS.LMC= 000007            OP$EVE= 010000            P$$WRD= 000000
D$NBEA  000056            LO.LDT  000046            LS.LOG= 000020            OP$FIL= 000200            QF$ACT= 000001
D$NBRA  000054            LO.LID  000030            LS.MOD= 000036            OP$GRP= 000100            QF$ALL= 000002
D$NEND= 000054            LO.LIN  000004            LS.NOD= 000014            OP$HOS= 000001            QF$KNO= 000004
D$NLN   000030            LO.LTM  000034            LS.NTI= 000013            OP$HTM= 000002            QF$LDO= 000010
D$NN    000060            LO.MAC  000022            LS.OBJ= 000022            OP$INA= 004000            Q$$OPT= 000010
D$OUTT  000043            LO.MDE  000024            LS.OPT= 000400            OP$INC= 001000            RF.LOO= 100000
D$RETF  000050            LO.NAM  000044            LS.PRO= 000002            OP$KDS= 002000            RQB   = ****** GX
D$RNN   000002            LO.NET= 000002            LS.TOP= 002000            OP$KDT= 000400            R$$DER= 000000
D$RTMR  000076            LO.NNM  000022            LS.TST= 000010            OP$KEV= 001000            R$$K11= 000001
D$SEG   000036            LO.OFL  000012            LS.UNF= 020000            OP$KGR= 001000            R$$SND= 000000
D$SER   000032            LO.ONM  000004            LS.XIT= 000011            OP$KNT= 010000            R$$11M= 000000
D$SQRL  000052            LO.OUT  000056            LX.ALI= 000017            OP$KSK= 002000            R.ADD   000010
D$$BUG= 177514            LO.OWN  000026            LX.CEX= 000004            OP$LCT= 000200            R.FLAG  000012
D$$ISK= 000000            LO.PAR  000010            LX.CIR= 000034            OP$LNE= 000040            R.LEN   000014
D$$L11= 000001            LO.PRI  000012            LX.LIN= 000006            OP$LOC= 000001            R.LNK   000000
D$$YNC= 000000            LO.RET  000064            LX.LOG= 000021            OP$LST= 000002            R.NAM   000002
D$$YNM= 000000            LO.RPA  000030            LX.NOD= 000015            OP$LTM= 000001            SYSFDB= ****** GX
E$$XPR= 000000            LO.SCR  000016            LX.OBJ= 000023            OP$LVL= 000010            S$$WRG= 000000
F$$LVL= 000001            LO.SEG  000054            LX.PRO= 000005            OP$MAC= 000010            S$$YSZ= 007600
```

```
                                  .SBTTL  LOCAL DATA

                          ;
                          ; LOCAL DATA FOR CUG$DF MACRO
                          ;
000000                            .PSECT  DATA,D
                                  .NLIST  BEX

000000                    CUGNML: .BLKB   1                 ; Length of CUG name                              ;[TD001]
000001                    CUGNAM: .BLKB   MAXID             ; Closed user group name (must follow CUGNML)     ;[TD001]
          000004                  CUGNLN = 4                ; MAXIMUM DIGITS IN CUG NUMBER                    ;**-1
000021       000          NUMPRE: .BYTE   0                 ; CUG NUMBER PREFIX IF LESS THAN 4 DIGITS
000022       000                  .BYTE   0                 ; ...
000023       000                  .BYTE   0                 ; ...
000024                    CUGNUM: .BLKB   4                 ; CLOSED USER GROUP NUMBER
          000027'                 CUGEND=.-1                ; END OF CUG NAME
000030                    CUGPCK: .BLKB   CUGNLN/2          ; CUG NUMBER IN BCD FORMAT
000032                    CUGFLG: .BLKB   1                 ; FLAGS BYTE
                                  .EVEN
```

```
VFPCUG - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:33  Page 18-2
Symbol table

ALCUG   000712R        D$$ISK= 000000         H$GPT   000046         M$$MGE= 000000         $BFR  = ****** GX
ALCUG0  000644R        D$$L11= 000001         H$HITS  000034         M$$NET= 000000         $BLANK= 000006
ALCUG1  000672R        D$$YNC= 000000         H$HLEN  000044         M$$OVR= 000000         $CEACC= ****** GX
A$$CHK= 000000         D$$YNM= 000000         H$LBDA  000070         NEXT  = ****** GX      $DIGIT= 000024
A$$CPS= 000000         END     000050R    003 H$LBDN  000072         NUMPRE  000021R    002 $DNUMB= 000014
A$$PRI= 000000         E$$XPR= 000000         H$LDTE  000002         N$$ACC= 000001         $EOS  = 000012
A$$TRP= 000000         FMT10 = ****** GX      H$LEN   000042         N$$BUF= 000001         $ERRM1  000126R    002
CERR  = ****** GX      FMT8  = ****** GX      H$LOTS  000032         N$$LDV= 000001         $ERRN2  000034R    002
CFGBF = ****** GX      FM.10 = 000000         H$NETW  000024         N$$MCP= 000001         $ERRN3  000056R    002
CFGSZ = ****** GX      FM.8  = 000000         H$NML = 000006         N$$MLL= 000001         $ERRP4  000102R    002
CFLIN = ****** GX      F$$LVL= 000001         H$NPT   000022         N$$MOV= 000010         $ERR1T= ****** GX
CKCUG0  000336R        F.RSIZ= ****** GX      H$PTB   000020         N$$NCT= 000001         $EXIT = 000000
CKCUG1  000504R        GF$BUG= 000001         H$PVC   000006         N$$PEM= 000001         $FAIL = 177777
COUNT = ****** GX      G$CUG   000010         H$RDTE  000004         PCKBCD= ****** GX      $GETAD= ****** GX
CUEND   000250R        G$DTE   000012         H$RNW   000042         PF.XDF= ****** GX      $GETRV= ****** GX
CUFG    000226R        G$FLG   000014         H$SVC   000036         P$$P45= 000000         $LAMDA= 000000
CUGDF   000000R    003 G$GNAM  000016         H$TRB   000016         P$$WRD= 000000         $NUMBR= 000002
CUGDIG  000142R        G$LEN   000016         H$XAVL  000100         Q$$OPT= 000010         $PFLAG= ****** GX
CUGEND= 000027R    002 G$LNK   000000         H$XBIA  000074         RTSPC = ****** GX      $PSIPT= ****** GX
CUGFLG  000032R    002 G$NAM   000002         H$X29C  000040         R$$DER= 000000         $PUTAD= ****** GX
CUGKW   000000RG   004 G$NML   000015         I$$RAR= 000000         R$$K11= 000001         $QCUG   000000RG
CUGNAM  000001R    002 G$$TPP= 000000         I$$RDN= 000000         R$$SND= 000000         $RADDR= ****** GX
CUGNLN= 000004         G$$TSS= 000000         K$$CNT= 177546         R$$11M= 000000         $RAD50= 000016
CUGNML  000000R    002 G$$TTK= 000000         K$$CSR= 177546         SPACE = 000040         $RBLCK= ****** GX
CUGNUM  000024R    002 G$$WRD= 000000         K$$LDC= 000000         STORID= ****** GX      $SAVAL= ****** GX
CUGPCK  000030R    002 HF$DLM= 000002         K$$TPS= 000074         STRNXT= ****** GX      $SAVRG= ****** GX
CUGST   000000RG   003 HF$GWY= 000010         LD$LP = 000000         SYNERR= ****** GX      $STRNG= 000004
CUNAM   000052R        HF$HOS= 000004         LNKEND= ****** GX      SYSFDB= ****** GX      $SUBXP= 000010
CUNBEN  000166R        HF$XDF= 000020         L$$ASG= 000000         S$$WRG= 000000         $$$FLG= 177777
CUNBST  000104R        H$CUG   000010         L$$DRV= 000000         S$$YSZ= 007600         $$$KEY= 000000
CUNMMX= 000006         H$DST   000012         L$$P11= 000001         T$$KMG= 000000          $$R  = 000010
CUNXT   000020R    003 H$D29   000014         L$$11R= 000000         T$$MIN= 000000         $$$STA= 000000
C$$CKP= 000000         H$FLG   000000         MAXID = 000020         V$$CTR= 001000         $$$TMP= 000000R    005
C$$ORE= 000400         H$GLEN  000104         MOVE  = ****** GX      X$$DBT= 000000         .PNUMB= ****** GX
C$$RSH= 177564         H$GLT   000044         M$$CRB= 000124         $ALPHA= 000022         .PNUMH= ****** GX
DTEDES= ****** GX      H$GNAM  000050         M$$CRX= 000000         $ALPOL= ****** GX      .TPARS= ****** GX
D$$BUG= 177514         H$GNML= 000020         M$$FCS= 000000         $ANY  = 000020

. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        001116    001   (RW,I,LCL,REL,CON)
DATA    000200    002   (RW,D,LCL,REL,CON)
$STATE  000060    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13706 Words  ( 54 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:33.59
SY:VFPCUG.V2,[132,134]VFPCUG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPCUG
```

```

                                           .SBTTL  DSA$DF ACTION ROUTINES

                                   ;
                                   ; START OF REMOTE DTE ADDRESS
                                   ;
 000060  012700  000000G           DARDST: MOV     #DARMLN,R0               ; GET LENGTH OF REMOTE DTE ADDRESS
 000064  012701  000000G                   MOV     #DSARDT,R1               ; GET ADDRESS OF START OF BUFFER
 000070  105021                    10$:    CLRB    (R1)+                    ; INITIALIZE BUFFER
 000072  005300                            DEC     R0                       ; MORE TO INITIALIZE?
 000074  003375                            BGT     10$                      ; BR IF YES
 000076  105067  000000G                   CLRB    COUNT                    ; INITIALIZE DIGIT COUNT
 000102  012767  000000G 000000G           MOV     #DSARDT,NEXT             ; START AT BEGINNING OF BUFFER
 000110                                    CALL    STRNXT                   ; STORE CHARACTER
 000114                                    RETURN

                                   ;
                                   ; PROCESS REMOTE DTE DIGIT
                                   ;
 000116  026727  000000G 000000G   DARDNO: CMP     NEXT,#DARDEN             ; IS ADDRESS TOO LONG?
 000124  101003                            BHI     101$                     ; BR IF YES
 000126                                    CALL    STRNXT                   ; STORE CHARACTER
 000132                                    RETURN
                                   ;
                                   ; ERRORS
                                   ;
 000134                            101$:   EMSG$R  N8                       ; ILLEGAL REMOTE DTE ADDRESS

                                   ;
                                   ; END OF REMOTE DTE ADDRESS
                                   ;
 000142  004567  000000G           DARDND: JSR     R5,$SAVRG                ; SAVE R3-R5
 000146  012700  000000G                   MOV     #<DARMLN/2>+1,R0         ; GET LENGTH OF PACKED ADDRESS
 000152  012704  000000G                   MOV     #DSARDT,R4               ; POINT TO NUMBER TO CONVERT
 000156  012705  000000G                   MOV     #DSARDP,R5               ; POINT TO STORAGE FOR PACKED NUMBER
 000162                                    CALL    PCKBCD                   ; PACK REMOTE DTE ADDRESS IN BCD
 000166  116767  000000G 000000G           MOVB    COUNT,DSARCT             ; STORE NUMBER OF DIGITS IN DTE ADDRESS
 000174                                    RETURN
```

```
     65                                                  .SBTTL  Macro definitions
     66
     67                                          ;
     68                                          ; LIBRARY MACROS
     69                                          ;
     70                                                  .MCALL  EMSG$R,NTLER$,STATE$,TRAN$,ISTAT$
     71                                                  .MCALL  VNPDBG
     72
     73 000000                                           .PSECT  DATA,D
     74                                                  .NLIST  BEX
     75
     76 000000                                           NTLER$  ,MC,10,CERR,RTSPC,CFLIN,<destination access data>
     77
     78                                                  .LIST   BEX
     79                                                  .EVEN
```

```
VFPDSN     CREATED BY  MACRO  ON 29-JUN-85 AT 02:35     PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE            REFERENCES

$NUMBR  = 000002          #9-122
$QDSN     000000 RG       #8-103
$RAD50  = 000016          #9-122
$RONLY  = ******           9-122      9-122      9-122
$STRNG  = 000004          #9-122
$SUBXP  = 000010          #9-122
$$$FLG  = 177777          #9-122
$$$KEY  = 177777          #9-122
.PCHAR  = ******   GX      16-336
.PSTCN  = ******   GX      11-227
.PSTPT  = ******   GX      11-231
.TPARS  = ******   GX      8-110
```

```
                                                .SBTTL  DSTEND - END OF DESTINATION DESCRIPTOR BLOCK MACROS

                                        ;
                                        ; DSTEND - END OF DESTINATION DESCRIPTOR BLOCK MACROS
                                        ;
000266                                  DSTEND::VNPDBG  DSTEND                  ; VNP debugging break point             ;[TD001]
000266  126767  000000G 000000G                 CMPB    DSCMCT,DSCVCT           ; ARE CALL DATA MASK/VALUE SAME LENGTH? ;[TD001]
000274  001043                                  BNE     104$                    ; BR IF NO - ERROR                      ;**-1
000276  016700  000000G                         MOV     $PSIPT,R0               ; POINT TO PSI HOME BLOCK
000302  132767  000000G 000000G                 BITB    #DS.X29,DSFLG           ; IS THIS AN X29 DESCRIPTOR BLOCK?
000310  001003                                  BNE     10$                     ; BR IF YES
000312  062700  000012                          ADD     #H$DST,R0               ; ELSE POINT TO X25 LISTHEAD
000316  000402                                  BR      20$                     ; CONTINUE IN COMMON CODE
000320  062700  000014          10$:            ADD     #H$D29,R0               ; POINT TO X29 LISTHEAD
000324  032767  000000G 000000G 20$:            BIT     #PF.XDF,$PFLAG          ; Check type of system                  ;[TD001]
000332  001407                                  BEQ     30$                                                             ;[TD001]
                                        ;                                                                               ;[TD001]
                                        ; Check and allocate destination blocks for extended-data-format system         ;[TD001]
                                        ;                                                                               ;[TD001]
000334                                          CALL    CKDSN0                  ; Check for unique destination name     ;[TD001]
000340  103413                                  BCS     101$                                                            ;[TD001]
000342                                          CALL    ALDST0                  ; Allocate and fill destination block   ;[TD001]
000346  103413                                  BCS     103$                                                            ;[TD001]
000350  000406                                  BR      50$                                                             ;[TD001]
                                        ;                                                                               ;[TD001]
                                        ; Check and allocate destination blocks for old-format systems                  ;[TD001]
                                        ;                                                                               ;[TD001]
000352                                  30$:    CALL    CKDSN1                  ; Check for unique destination name     ;[TD001]
000356  103404                                  BCS     101$                                                            ;[TD001]
000360                                          CALL    ALDST1                  ; Allocate and fill destination block   ;[TD001]
000364  103404                                  BCS     103$                                                            ;[TD001]
                                        ;                                                                               ;[TD001]
000366                                  50$:    RETURN                                                                  ;[TD001]
                                        ;                                                                               ;**-15
                                        ; ERRORS
                                        ;
000370                                  101$:   EMSG$R  M7                      ; DESTINATION NAME NOT UNIQUE
000376                                  103$:   EMSG$R  M9                      ; DESTINATION BLOCK ALLOCATION FAILURE
000404                                  104$:   EMSG$R  6M                      ; ILLEGAL CALL DATA MASK/VALUE
```

```
VFPDST - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:35  Page 23-1
Symbol table

ALDST0  000654R         DSTVR1= ****** GX       HF$HOS= 000004          MOVBCD  001704R         $ALPHA= 000022
ALDST1  001152R         DS.X25= ****** GX       HF$XDF= 000020          MOVE  = ****** GX       $ALPOL= ****** GX
A$$CHK= 000000          DS.X29= ****** GX       H$CUG   000010          MOVLEN  001732R         $ANY  = 000020
A$$CPS= 000000          D$ACL   000027          H$DST   000012          MXNOID= 000006          $BFR  = ****** GX
A$$PRI= 000000          D$CUGL  000023          H$D29   000014          M$$CRB= 000124          $BLANK= 000006
A$$TRP= 000000          D$CUGN  000022          H$FLG   000000          M$$CRX= 000000          $CAT5B= ****** GX
BCDLEN  001642R         D$DATL  000030          H$GLEN  000104          M$$FCS= 000000          $CEACC= ****** GX
CERR  = ****** GX       D$DSL   000022          H$GLT   000044          M$$MGE= 000000          $DIGIT= 000024
CFGBF = ****** GX       D$DST   000010          H$GNAM  000050          M$$NET= 000000          $DNUMB= 000014
CFGSZ = ****** GX       D$DTEL  000031          H$GNML= 000020          M$$OVR= 000000          $EOS  = 000012
CFLIN = ****** GX       D$FLEN  000032          H$GPT   000046          N$$ACC= 000001          $ERRM7  000204R     002
CKDSN0  000412R         D$GLEN  000032          H$HITS  000034          N$$BUF= 000001          $ERRM9  000236R     002
CKDSN1  000534R         D$GVBL  000032          H$HLEN  000044          N$$LDV= 000001          $ERRN4  000000R     002
CUNMMX= 000006          D$LNK   000000          H$LBDA  000070          N$$MCP= 000001          $ERRN5  000032R     002
C$$CKP= 000000          D$NAM   000004          H$LBDN  000072          N$$MLL= 000001          $ERRN6  000062R     002
C$$ORE= 000400          D$NOD   000010          H$LDTE  000002          N$$MOV= 000010          $ERRN7  000110R     002
C$$RSH= 177564          D$OBJ   000003          H$LEN   000042          N$$NCT= 000001          $ERR1T= ****** GX
DSACGL= ****** GX       D$PAL   000026          H$LOTS  000032          N$$PEM= 000001          $ERR6M  000132R     002
DSACUG= ****** GX       D$PRI   000002          H$NETW  000024          PF.XDF= ****** GX       $EXIT = 000000
DSARCT= ****** GX       D$PRL   000024          H$NML = 000006          P$$P45= 000000          $FAIL = 177777
DSASHI= ****** GX       D$SHI   000020          H$NPT   000022          P$$WRD= 000000          $GETAD= ****** GX
DSASLO= ****** GX       D$SLO   000016          H$PTB   000020          Q$$OPT= 000010          $GETRV= ****** GX
DSCMCT= ****** GX       D$USL   000025          H$PVC   000006          RNAL  = ****** GX       $LAMDA= 000000
DSCVCT= ****** GX       D$VBL   000032          H$RDTE  000004          RNNL  = ****** GX       $NUMBR= 000002
DSFLG = ****** GX       D$$BUG= 177514          H$RNW   000042          RNNODE= ****** GX       $PFLAG= ****** GX
DSNAM   000056R         D$$ISK= 000000          H$SVC   000036          RNPL  = ****** GX       $PSIPT= ****** GX
DSNAMX= 000006          D$$L11= 000001          H$TRB   000016          RNUL  = ****** GX       $PUTAD= ****** GX
DSNM    000024R     003 D$$YNC= 000000          H$XAVL  000100          RTSPC = ****** GX       $QDST   000000RG
DSOBJ   000170R         D$$YNM= 000000          H$XBIA  000074          R$$DER= 000000          $RADDR= ****** GX
DSPRI   000162R         END     000062R     003 H$X29C  000040          R$$K11= 000001          $RAD50= 000016
DSTDF   000000R     003 E$$XPR= 000000          I$$RAR= 000000          R$$SND= 000000          $RBLCK= ****** GX
DSTEND  000266RG        FMT10 = ****** GX       I$$RDN= 000000          R$$11M= 000000          $SAVAL= ****** GX
DSTINS  001504R         FMT8  = ****** GX       K$$CNT= 177546          SPACE = 000040          $STRNG= 000004
DSTKMX= 000006          FM.10 = 000000          K$$CSR= 177546          SPFILL= ****** GX       $SUBXP= 000010
DSTKW   000000RG    004 FM.8  = 000000          K$$LDC= 000000          STORID= ****** GX       $XLINK= ****** GX
DSTNAM= ****** GX       FNDLEN  001626R         K$$TPS= 000074          SYNERR= ****** GX       $$$FLG= 177777
DSTNML= ****** GX       F$$LVL= 000001          LD$LP = 000000          SYSFDB= ****** GX       $$$KEY= 000002
DSTOBJ= ****** GX       F.RSIZ= ****** GX       L$$ASG= 000000          S$$WRG= 000000          $$$R  = 000010
DSTPRI= ****** GX       G$$TPP= 000000          L$$DRV= 000000          S$$YSZ= 007600          $$$STA= 000000
DSTSK   000240R         G$$TSS= 000000          L$$P11= 000001          T$$KMG= 000000          $$$TMP= 000013R     005
DSTST   000000RG    003 G$$TTK= 000000          L$$11R= 000000          T$$MIN= 000000          .PNUMB= ****** GX
DSTTKL= ****** GX       G$$WRD= 000000          MAXID = 000020          V$$CTR= 001000          .PNUMH= ****** GX
DSTTSK= ****** GX       HF$DLM= 000002          MOVAR   001662R         X$$DBT= 000000          .TPARS= ****** GX
DSTVR0= ****** GX       HF$GWY= 000010

. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        001744    001   (RW,I,LCL,REL,CON)
DATA    000314    002   (RW,D,LCL,REL,CON)
$STATE  000072    003   (RW,D,LCL,REL,CON)
$KTAB   000006    004   (RW,D,LCL,REL,CON)
$KSTR   000017    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
```

```
                                         .SBTTL  LOOK FOR DTE$DF MACRO

                                 ;+
                                 ; $QDTE - LOOK FOR DTE$DF MACRO
                                 ;
                                 ; INPUTS:
                                 ;       NONE
                                 ;
                                 ; OUTPUTS:
                                 ;       C-BIT=SUCCESS/FAILURE
                                 ;       R3,R4,R5=DESTROYED
                                 ;-
                                                                                                   ;[TD001]
 000000                          $QDTE:: VNPDBG  VFPDTE          ; VNP debugging break point       ;[TD001]
 000000  012705  000000'                 MOV     #DTEDF,R5       ; STATE TABLE ADDRESS             ;[TD001]
 000004  005001                          CLR     R1              ; FULL KEYWORD MATCH LENGTH       ;**-1
 000006  012702  000000'                 MOV     #DTEKW,R2       ; KEYWORD TABLE ADDRESS
 000012  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
 000016  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
 000022  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
 000026                                  CALL    .TPARS          ; GO DO THE PARSE
 000032  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
 000034  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
 000040  001401                          BEQ     101$            ; IF EQ, YES
 000042                          20$:    RETURN

                                 ;
                                 ; ERROR CONDITION
                                 ;
 000044                          101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
VFPDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:37      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

BIASX      001322 R          15-413     #18-537
CERR     = ******    GX       8-125      8-126      8-127      8-128      8-129      8-130
CFGBF    = ******    GX       9-153
CFGSZ    = ******    GX       9-152
CFLIN    = ******    GX       8-125      8-126      8-127      8-128      8-129      8-130
CHKDEV     001206 R          15-355     #17-491
CHKDTE     001042 R          15-350     #16-446
CNTTIM     000326 R         #13-317
COUNT    = ******    GX     *12-251      15-385
CTIM     = ******    GX     *13-319      15-400
DE.ON    = ******    GX      15-403
DTEADD     000000 R         #7-115       12-247     12-252     12-273
DTEAEN     000140 R         #12-271
DTEALN   = 000017           #7-114       7-115      12-246     12-272     15-380     16-458
DTEAST     000052 R         #12-245
DTEDES   = ******    GX     *15-371      15-419
DTEDF      000000 R          9-149
DTEDIG     000114 R         #12-259
DTEEND     000412 R          14-339     #15-350
DTEFLG   = ******    GX     *12-245      15-403
DTEKW      000000 RG         9-151      #10-170
DTEND    = 000016 R         #7-116       12-259
DTENET     000352 R         #14-331
DTEPCK     000017 R         #7-117       12-274     15-381     16-461
DTEST      000000 RG        #10-170
DTNMMX   = 000006           #7-118       7-119      12-278     12-287     14-332     15-389
DTNTNM     000027 R         #7-119       12-277     12-286     14-335     15-390
FMT10    = ******    GX       8-125      8-126      8-128
FMT8     = ******    GX       8-127      8-129      8-130
FM.10    = 000000           #8-125      #8-126     #8-128
FM.8     = 000000           #8-127      #8-129     #8-130
FNDPDV   = ******    GX      17-497
F.RSIZ   = ******    GX     *12-284     *15-416    *16-449    *16-457
HSHADD   = ******    GX     *15-412
HSHMN    = 000040           #6-87        13-300
HSHMX    = 001000           #6-88        13-304
HSHSZ    = ******    GX     *13-307      15-358     15-415
HSHTBL     00C256 R         #13-297
H$LDTE     000002            15-418      16-448
H$LEN      000042            12-284
H$NETW     000024            12-285
LF.ACT   = 100000           #5-69
LF.BRO   = 000400           #5-69
LF.BWT   = 000007           #5-69
LF.ENA   = 002000           #5-69
LF.LPB   = 001000           #5-69
LF.MDC   = 000100           #5-69
LF.MFL   = 004000           #5-69
LF.MTP   = 000020           #5-69        17-516
LF.PAC   = 000200           #5-69
LF.RDY   = 040000           #5-69
LF.REA   = 010000           #5-69
```

```
    150                                           .SBTTL  TPARS STATE TABLES
    151                                   ;
    152                                   ; TPARS STATE TABLES
    153                                   ;
    154 000052                                    ISTAT$  PSNST,PSNKW
    155
    156                                   ;
    157                                   ; PSN$DF
    158                                   ;
    159 000052                                    STATE$  PSNDF
    160 000052                                     TRAN$   %PSN$DF%,,,1,SYNERR
    161
    162 000052                                    STATE$
    163 000052                                     TRAN$   $RAD50,,PSNNM          ; NETWORK NAME                    ;[TD001]
    164                                                                                                              ;**-1
    165 000052                                    STATE$
    166 000052                                     TRAN$   <',>
    167
    168 000052                                    STATE$
    169 000052                                     TRAN$   $NUMBR,,PORT           ; NUMBER OF PORTS SUPPORTED
    170
    171 000052                                    STATE$
    172 000052                                     TRAN$   !END,$EXIT,PSNEND
    173 000052                                     TRAN$   <',>
    174
    175 000052                                    STATE$
    176 000052                                     TRAN$   <',>,PSLO
    177 000052                                     TRAN$   $NUMBR,,PSNFG          ; FLAGS
    178
    179 000052                                    STATE$
    180 000052                                     TRAN$   !END,$EXIT,PSNEND
    181 000052                                     TRAN$   <',>
    182
    183 000052                                    STATE$  PSLO
    184 000052                                     TRAN$   <',>,PSHI
    185 000052                                     TRAN$   $NUMBR,,PSNSLO         ; LOW END OF SUBADDRESS RANGE
    186
    187 000052                                    STATE$
    188 000052                                     TRAN$   !END,$EXIT,PSNEND
    189 000052                                     TRAN$   <',>
    190
    191 000052                                    STATE$  PSHI
    192 000052                                     TRAN$   $NUMBR,END,PSNSHI      ; HIGH END OF SUBADDRESS RANGE
    193
    194 000052                                    STATE$  END                    ; END OF STRING SUB-EXPRESSION
    195 000052                                     TRAN$   $EOS,$EXIT
    196 000052                                     TRAN$   <';>,$EXIT
    197
    198 000052                                    STATE$
```

```
**FILE**ID**VFPPVC











VV      VV   FFFFFFFFFF   PPPPPPPP     PPPPPPPP     VV      VV     CCCCCCCC
VV      VV   FFFFFFFFFF   PPPPPPPP     PPPPPPPP     VV      VV     CCCCCCCC
VV      VV   FF           PP      PP   PP      PP   VV      VV   CC
VV      VV   FF           PP      PP   PP      PP   VV      VV   CC
VV      VV   FF           PP      PP   PP      PP   VV      VV   CC
VV      VV   FF           PP      PP   PP      PP   VV      VV   CC
VV      VV   FFFFFFFF     PPPPPPPP     PPPPPPPP     VV      VV   CC
VV      VV   FFFFFFFF     PPPPPPPP     PPPPPPPP     VV      VV   CC
VV      VV   FF           PP           PP           VV      VV   CC
VV      VV   FF           PP           PP           VV      VV   CC
  VV  VV     FF           PP           PP             VV  VV     CC               ....
  VV  VV     FF           PP           PP             VV  VV     CC               ....
    VV       FF           PP           PP               VV         CCCCCCCC       ....
    VV       FF           PP           PP               VV         CCCCCCCC       ....






LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                                  F
VFGSVC - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 02:30  Page 17-1
Symbol table

A$$CHK= 000000        HTIMR   000002R   002 LN.DN = 000000        N$TCTL  000112        PX$DLM= 000200
A$$CPS= 000000        H$CUG   000010        LN.ODP= 000004        N$TLC   000100        PX$SVC= 000100
A$$PRI= 000000        H$DST   000012        LN.DPE= 000001        N$TNC   000114        P$CHN   000004
A$$TRP= 000000        H$D29   000014        LN.REF= 000002        N$TRC   000106        P$CNT   000005
CFERR = ****** GX     H$FLG   000000        LN.SER= 000002        N$TTCB  000110        P$CTR   000034
CFGBF = ****** GX     H$GLEN  000104        LN.STA= 000017        N$VER   000066        P$DRCT  000015
CFGSZ = ****** GX     H$GLT   000044        LN.SUB= 000360        N$XLEN  000124        P$DRTR  000036
COUNT   000024R   002 H$GNAM  000050        LN.TRI= 000006        N$$ACC= 000001        P$FLG   000012
CTLSVC  000021R   002 H$GNML= 000020        LTIMR   000004R   002 N$$BUF= 000001        P$FWD   000030
CVTBUF  000054R   002 H$GPT   000046        L$$ASG= 000000        N$$LDV= 000001        P$ICCB  000046
C$$CKP= 000000        H$HITS  000034        L$$DRV= 000000        N$$MCP= 000001        P$IPL   000014
C$$ORE= 000400        H$HLEN  000044        L$$P11= 000001        N$$MLL= 000001        P$LCD   000002
C$$RSH= 177564        H$LBDA  000070        L$$11R= 000000        N$$MOV= 000010        P$LEN   000052
DTEEND= 000044R   002 H$LBDN  000072        L.COST  000015        N$$NCT= 000001        P$LST   000000
DTELEN= 000017        H$LDTE  000002        L.CTL   000012        N$$PEM= 000001        P$NRNI  000040
DTEPCK  000044R   002 H$LEN   000042        L.CVA   177776        OWNSVC  000020R   002 P$OCCB  000050
DTESVC  000025R   002 H$LOTS  000032        L.DDM   000002        PCKBCD  001566R       P$PFQ   000006
D$$BUG= 177514        H$NETW  000024        L.DDS   000004        PF$BLK= 020000        P$PKSZ  000044
D$$ISK= 000000        H$NML = 000006        L.DLC   000003        PF$CLC= 010000        P$RMX1  000016
D$$L11= 000001        H$NPT   000022        L.DLM   000006        PF$DLM= 100000        P$RMX2  000020
D$$YNC= 000000        H$PTB   000020        L.DLS   000010        PF$EIP= 000002        P$RPRI  000042
D$$YNM= 000000        H$PVC   000006        L.FLG   000000        PF$ENB= 000001        P$RTIM  000003
E$NBR   000014        H$RDTE  000004        L.KRBA  000016        PF$FAI= 004000        P$STA1  000022
E$NBS   000020        H$RNW   000042        L.LEN = 000022        PF$FM1= 000100        P$STA2  000023
E$NCR   000034        H$SVC   000036        L.MPF   000022        PF$FM2= 000200        P$TIM   000010
E$NCS   000036        H$TRB   000016        L.NMST  000020        PF$DFF= 000000        P$TSCT  000026
E$NIC   000044        H$XAVL  000100        L.NSTA  000014        PF$RM1= 000020        P$TSIZ  000024
E$NLEN  000050        H$XBIA  000074        L.OWNR  000021        PF$RM2= 000040        P$TYP   000001
E$NLLA  000012        H$X29C  000040        L.UNT   000013        PF$RVR= 000010        P$$P45= 000000
E$NLNK  000000        I$$RAR= 000000        MAXWND= 000007        PF$STA= 000004        P$$WRD= 000000
E$NML   000040        I$$RDN= 000000        M$$CRB= 000124        PF$SVC= 040000        Q$$DPT= 000010
E$NMR   000024        K$$CNT= 177546        M$$CRX= 000000        PF$UP = 000004        RCLSVC  000023R   002
E$NMS   000030        K$$CSR= 177546        M$$FCS= 000000        PKTSVC  000006R   002 RF.CTL= 000003
E$NNOD  000002        K$$LDC= 000000        M$$MGE= 000000        PR$BED= 000200        RF.LD1= 000040
E$NRT   000042        K$$TPS= 000074        M$$NET= 000000        PR$BEU= 000100        RF.LD2= 000100
E$NRTP  000005        LD$LP = 000000        M$$OVR= 000000        PR$BRD= 000040        RF.TIM= 177400
E$NSEG  000010        LF.ACT= 100000        NEXT    000000R   002 PR$BRU= 000020        RF.TMO= 000400
E$NTIM  000046        LF.BRD= 000400        NODTE   000420R       PR$DWN= 000002        RF.WFC= 000200
E$NUSE  000004        LF.BWT= 000007        N$ADJ1  000072        PR$LCC= 000010        RF.WTD= 000020
E$STRT  000006        LF.ENA= 002000        N$ADJ2  000074        PR$MOP= 000004        RF.WTM= 000030
E$$XPR= 000000        LF.LPB= 001000        N$CACH  000062        PR$UP = 000001        RF.WTS= 000010
FLGSVC  000010R   002 LF.MDC= 000100        N$CRC   000120        PS$CHR= 000016        RTMSVC  000012R   002
FMT8  = ****** GX     LF.MFL= 004000        N$HC1   000052        PS$FAI= 000014        RTSPC = ****** GX
FMT8B = ****** GX     LF.MTP= 000020        N$HC2   000056        PS$NTI= 000006        R$$DER= 000000
FM.8  = 000000        LF.PAC= 000200        N$LV1   000002        PS$OFF= 000000        R$$K11= 000001
FM.8B = 000000        LF.RDY= 040000        N$LV2   000010        PS$STR= 000002        R$$SND= 000000
FNDOWN  001214R       LF.REA= 010000        N$MHC1  000036        PS$UP = 000012        R$$11M= 000000
F$$LVL= 000001        LF.SER= 000040        N$MHC2  000044        PS$VER= 000010        SETTLC  001272R
F.RSIZ= ****** GX     LF.TIM= 000010        N$PLD   000016        PS$WT = 000004        SF$BLK= 000001
G$$TPP= 000000        LF.UNL= 020000        N$PRI   000076        PT$BRD= 000200        SF$INC= 000002
G$$TSS= 000000        LF.X2P= 000000        N$ROA1  000022        PT$DRT= 000100        SF.ACT= 000200
G$$TTK= 000000        LN.CLO= 000000        N$ROA2  000030        PT$END= 000004        SF.ENA= 000100
G$$WRD= 000000        LN.DUM= 000005        N$RTMX  000014        PT$LV1= 000002        SF.LPB= 000004
HF$DLM= 000002        LN.LOA= 000004        N$RTM1  000014        PT$LV2= 000001        SF.MFL= 000040
HF$GWY= 000010        LN.LOD= 000003        N$RTM2  000015        PT$PH3= 000010        SF.PAC= 000020
HF$HOS= 000004        LN.OAU= 000003        N$RT1   000000        PT$XAR= 000020        SF.REA= 000010
HF$XDF= 000020        LN.OFF= 000001        N$RT2   000006        PX$BLK= 000040        SF.SER= 000001
```

```
                                                                  G 1
VFGSVC - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 02:30  Page 17-2
```

```
     82                                          .SBTTL  LOCAL DATA
     83
     84                                  ;
     85                                  ; LOCAL DATA
     86                                  ;
     87 000000                           X2DEV:  .BLKW   1                               ; DEVICE NAME
     88 000002                           X2CTL:  .BLKB   1                               ; CONTROLLER NUMBER
     89 000003                           X2UNT:  .BLKB   1                               ; UNIT NUMBER
     90
```

```
VFGX2P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32      PAGE 5
MACRO CROSS REFERENCE                                    CREF   04.00
MACRO NAME       REFERENCES

CALL              9-122      11-204
DBGTP$           #10-136    #10-154    #10-191
EMSG$R           #5-60       9-130      11-238     11-252     11-261     11-271     11-283     11-295
ISTAT$           #5-60       10-136
MTRAN$           #10-136
NTLER$           #5-60       8-98       8-99       8-100
PDVDF$           #5-60       5-62
REJ$             #5-67       11-235
RETURN            9-126      11-206     11-216     11-236     11-251     11-260     11-270     11-282     11-294
SLTDF$           #5-60       5-63
STATE$           #5-60       10-140    #10-143    #10-146    #10-149    #10-152    #10-156    #10-159    #10-162    #10-165
                 #10-168    #10-171    #10-174    #10-177    #10-180    #10-183    #10-186    #10-189    #10-193
 TRAN$           #5-60      #10-141    #10-144    #10-147    #10-150    #10-153    #10-154    #10-157    #10-160    #10-163
                 #10-166    #10-169    #10-172    #10-175    #10-178    #10-181    #10-184    #10-187    #10-190    #10-191
```

FILEID**VFGX3P

```
VFGX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE           REFERENCES

BLKSMN  = 000040        #6-67
BLKSMX  = 002000        #6-68
CERR    = ******  GX     8-91        8-92        8-93        8-94        8-95
CFGBF   = ******  GX     9-117
CFGSZ   = ******  GX     9-116
CFLIN   = ******  GX     8-91        8-92        8-93        8-94        8-95
CHKBLK  = ******  GX     11-181      11-189
CHKWND  = ******  GX     11-200      11-208
FMT8    = ******  GX     8-92        8-94
FMT8B   = ******  GX     8-91        8-93        8-95
FM.8    = 000000        #8-92       #8-94
FM.8B   = 000000        #8-91       #8-93       #8-95
NEXT      000000 R      #7-78       *11-219      11-231     *11-232
RTSPC   = ******  GX     8-91        8-92        8-93        8-94        8-95
SYNERR  = ******  GX    *9-118      *9-121      *11-234
S$$BAS  = ******         8-91        8-91        8-92        8-92        8-93        8-93        8-94        8-94        8-95
                         8-95
TIMRCT    000010 R      #7-82       *11-220     *11-224      11-225      11-241
WNDSMX  = 000177        #6-69
X3CTL     000004 R      #7-80
X3DEV     000002 R      #7-79
X3PDF     000000 R       9-113
X3PKW     000000 RG      9-115      #10-133
X3PST     000000 RG     #10-133
X3TMCT  = 000007        #6-70        11-225      11-241
X3UNT     000006 R      #7-81
X3.BEG    000212 R      #11-219
X3.DBS    000052 R      #11-181
X3.DWS    000132 R      #11-200
X3.END    000312 R      #11-241
X3.MBS    000074 R      #11-189
X3.MWS    000154 R      #11-208
X3.NXT    000224 R      #11-224
$ALPHA  = 000022        #10-133
$ANY    = 000020        #10-133
$BLANK  = 000006        #10-133
$DIGIT  = 000024        #10-133
$DNUMB  = 000014        #10-133
$EOS    = 000012        #10-133
$ERR1T  = ******  GX     9-127
$ERR2B    000012 R      #8-91        11-185
$ERR2C  = ******  GX     11-195
$ERR2D    000046 R      #8-92        11-196
$ERR2E    000122 R      #8-93        11 204
$ERR2F  = ******  GX     11-214
$ERR2G    000156 R      #8-94        11-215
$ERR2H    000232 R      #8-95        11-237
$EXIT   = 000000        #10-133
$FAIL   = 177777        #10-133
$GPRM   = ******         10-133
$LAMDA  = 000000        #10-133
$NUMBR  = 000002        #10-133
```

```
VFIND - VNP FIND ENTITY ROUTINE MACRO V05.03b Saturday 29-Jun-85 02:33  Page 10-2
Symbol table

TY$CHA= 000400          T$$MIN= 000000          WC.UNT= 001000          $FNADD  000330RG        $GETRV= ****** GX
TY$EVE= 001000          V$$CTR= 001000          X$$DBT= 000000          $FNALI  000116RG        $HMBUF= ****** GX
TY$STA= 002000          WC.CNT= 000400          $BFR  = ****** GX       $FNNAM  000470RG        $OBJHD= ****** GX
TY$SUM= 000000          WC.EVE= 004000          $CEACC= ****** GX       $FNOBJ  000002RG        $QUAL = ****** GX
T$$KMG= 000000          WC.TRI= 002000          $DECPT= ****** GX       $GETAD= ****** GX       $RADDR= ****** GX


. ABS.  000106    000   (RW,I,GBL,ABS,OVR)
        000652    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 15585 Words  ( 61 Pages)
Size of core pool: 16552 Words  ( 63 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:16.06
SY:VFIND.V2,[132,134]VFIND/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFIND
```

```
    109                                         .SBTTL  ERROR MESSAGES
    110
    111 000034                                  NTLER$  ,N2,10,CERR,RTSPC,CFLIN,<CUG name>                                           ;**-2
    112 000056                                  NTLER$  ,N3,10,CERR,RTSPC,CFLIN,<CUG number>
    113 000102                                  NTLER$  ,P4,10,CERR,RTSPC,CFLIN,<flags word>
    114 000126                                  NTLER$  ,M1,8,CERR,RTSPC,CFLIN,<CUG name block allocation failure>
    115
    116                                         .LIST   BEX                                                                          ;**-1
    117                                         .EVEN
    118 000000                                  .PSECT
```

```
VFPCUG     CREATED BY  MACRO  ON 29-JUN-85 AT 02:33      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

ALCUG      000712 R          16-377     17-395     #18-413
ALCUG0     000644 R          13-259    #16-372
ALCUG1     000672 R          13-267    #17-391
CERR    =  ******   GX       8-111      8-112      8-113      8-114
CFGBF   =  ******   GX       9-138
CFGSZ   =  ******   GX       9-137
CFLIN   =  ******   GX       8-111      8-112      8-113      8-114
CKCUG0     000336 R          13-257    #14-294
CKCUG1     000504 R          13-265    #15-334
COUNT   =  ******   GX      *11-213     12-235    *12-238
CUEND      000250 R         #13-251
CUFG       000226 R         #13-246
CUGDF      000000 R          9-134
CUGDIG     000142 R         #11-221
CUGEND  =  000027 R         #7-104      11-221
CUGFLG     000032 R         #7-106     *11-198    *13-250     18-435
CUGKW      000000 RG         9-136     #10-153
CUGNAM     000001 R         #7-98       14-309     15-347     18-437
CUGNLN  =  000004           #7-99       7-105      11-208     12-234     18-429
CUGNML     000000 R         #7-97       11-194     14-306     16-375     18-438     18-445
CUGNUM     000024 R         #7-103      11-209     11-214     12-236
CUGPCK     000030 R         #7-105      12-237     18-431
CUGST      000000 RG        #10-153
CUNAM      000052 R         #11-193
CUNBEN     000166 R         #12-233
CUNBST     000104 R         #11-208
CUNMMX  =  000006           #6-86       11-195     15-346
DTEDES  =  ******   GX       14-304     15-344     18-434
FMT10   =  ******   GX       8-111      8-112      8-113
FMT8    =  ******   GX       8-114
FM.10   =  000000           #8-111     #8-112     #8-113
FM.8    =  000000           #8-114
F.RSIZ  =  ******   GX      *14-297    *14-303    *15-337    *15-343    *18-423
G$CUG      000010            18-430
G$DTE      000012            14-304     15-344    *18-434
G$FLG      000014           *18-435
G$GNAM     000016            14-310     16-374
G$LEN      000016            14-303     15-343     16-376     17-394
G$NAM      000002            15-348     17-393
G$NML      000015            14-307     16-357     16-373
H$CUG      000010            14-296     15-336     18-449
LNKEND  =  ******   GX       18-450
MAXID   =  000020           #6-87       7-98       14-303
MOVE    =  ******   GX       18-432     18-441
NEXT    =  ******   GX      *11-214     11-221
NUMPRE     000021 R         #7-100
N$$VCT  =  ******            14-302     15-342     18-422
PCKBCD  =  ******   GX       12-239
PF.XDF  =  ******   GX       13-252
RTSPC   =  ******   GX       8-111      8-112      8-113      8-114
SPACE   =  000040           #6-85
STORID  =  ******   GX       11-196
```

```
                                 ;
                                 ; LOW END OF SUBADDRESS RANGE
                                 ;
000176                           DASSLO: CALL    CHKSB                   ; CHECK SUBADDRESS
000202  103405                           BCS     101$                    ; BR IF ILLEGAL
000204  010167  000000G                  MOV     R1,DSASLO               ; STORE LOW SUBADDRESS RANGE
000210  010167  000000G                  MOV     R1,DSASHI               ; DEFAULT HIGH SUBADDRESS IF OMITTED
000214                                   RETURN
                                 ;
                                 ; ERRORS
                                 ;
000216                           101$:   EMSG$R  N9                      ; ILLEGAL LOW SUBADDRESS



                                 ;
                                 ; START OF HIGH SUBADDRESS
                                 ;
000224                           DASSHI: CALL    CHKSB                   ; CHECK SUBADDRESS
000230  103406                           BCS     101$                    ; BR IF ILLEGAL
000232  020167  000000G                  CMP     R1,DSASLO               ; CHECK RANGE VALID
000236  002403                           BLT     101$                    ; BR IF ILLEGAL: HIGH LESS THAN LOW
000240  010167  000000G                  MOV     R1,DSASHI               ; STORE HIGH END OF SUBADDRESS RANGE
000244                                   RETURN
                                 ;
                                 ; ERRORS
                                 ;
000246                           101$:   EMSG$R  M0                      ; ILLEGAL HIGH SUBADDRESS


                                 ;
                                 ; ROUTINE TO CHECK SUBADDRESS
                                 ;
000254  005767  000000G          CHKSB:  TST     .PNUMH                  ; TOO BIG?
000260  001010                           BNE     20$                     ; BR IF YES
000262  016701  000000G                  MOV     .PNUMB,R1               ; GET SUBADDRESS
000266  002405                           BLT     20$                     ; BR IF ILLEGAL
000270  020127  000000G                  CMP     R1,#SUBAMX              ; NUMBER WITHIN LEGAL RANGE?
000274  101002                           BHI     20$                     ; BR IF ILLEGAL
000276  000241                           CLC                             ; ELSE SHOW NO ERROR
000300  000401                           BR      30$                     ; RETURN
000302  000261                   20$:    SEC                             ; FLAG ERROR
000304                           30$:    RETURN                          ;
```

```
     81                                  .SBTTL  Local Data
     82
     83 000040                  STRLEN: .BLKB   1                      ; Length of parsed string
     84 000041                  STRBUF: .BLKB   MAXSTR                 ; Buffer for parsed string
     85
     86                                  .EVEN
     87
     88 000000                           .PSECT
```

```
VFPDSN     CREATED BY  MACRO  ON 29-JUN-85 AT 02:35     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

CALL               8-110      11-233      12-257      13-280      14-304
DPGTP$           #9-122      #9-136      #9-142      #9-148      #9-168      #9-172      #10-184     #10-185     #10-189     #10-192
                 #10-193     #10-201     #10-204     #10-208     #10-209
EMSG$R           #6-70         8-118      11-236      12-260      13-283      14-307
ISTAT$           #6-70         9-122
MTRAN$           #9-122
NTLER$           #6-70         6-76
RETURN             8-114      11-234      12-258      13-281      14-305      15-320      16-338
STATE$           #6-70         9-124     #9-126      #9-129      #9-131      #9-134      #9-137      #9-140      #9-143      #9-146
                 #9-149      #9-152      #9-157      #9-159      #9-161      #9-163      #9-166      #9-170      #10-182     #10-187
                 #10-190     #10-194     #10-196     #10-199     #10-202     #10-206     #10-211
TRAN$            #6-70       #9-125      #9-127      #9-130      #9-132      #9-135      #9-136      #9-138      #9-141      #9-142
                 #9-144      #9-147      #9-148      #9-150      #9-153      #9-158      #9-160      #9-162      #9-164      #9-167
                 #9-168      #9-171      #9-172      #10-183     #10-184     #10-185     #10-188     #10-189     #10-191     #10-192
                 #10-193     #10-195     #10-197     #10-200     #10-201     #10-203     #10-204     #10-207     #10-208     #10-209
VNPDBG           #6-71         8-103
```

```
                                            .SBTTL  CKDSNO - Check destination name is unique [new format]                  ;[TD001]
                                 ;+                                                                                          ;[TD001]
                                 ;                                                                                           ;[TD001]
                                 ; *** CKDSNO - Check destination name is unique [new format]                                ;[TD001]
                                 ;                                                                                           ;**-5
                                 ; INPUTS:
                                 ;          R0 - ADDRESS OF DESTINATION DESCRIPTOR LISTHEAD
                                 ;
                                 ; OUTPUTS:
                                 ;          CARRY CLEAR - DESTINATION NAME IS UNIQUE
                                 ;          CARRY SET   - DESTINATION NAME IS NOT UNIQUE
                                 ;          Registers R3-R5 preserved                                                        ;[TD001]
                                 ;                                                                                           ;[TD001]
                                 ; *** REQUIRES DESTINATION NAME TO BE FIRST VARIABLE FIELD ***                              ;[TD001]
                                 ;-                                                                                          ;[TD001]
000412                           CKDSNO:                                                                                     ;[TD001]
000412                                      SAVMAP                          ; SAVE CURRENT MAPPING                           ;**-3
000416                                      GETRV   R0,#2                   ; READ IN ADDRESS LISTHEAD
000434  016700  000000G          10$:       MOV     $BFR,R0                 ; GET ADDRESS OF NEXT BLOCK
000440  000241                              CLC                             ; ASSUME SUCCESS
000442  001431                              BEQ     30$                     ; BR IF END OF LIST
000444                                      CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
000454                                      GETAD   R0,#D$GVBL+MAXID        ; Read fixed part and destination name           ;[TD001]
000472  116701  000022G                     MOVB    $BFR+D$DSL,R1           ; Get length of destination name                 ;[TD001]
000476  120167  000000G                     CMPB    R1,DSTNML               ; Check lengths                                  ;[TD001]
000502  001354                              BNE     10$                     ; If not equal then different names              ;[TD001]
000504  012702  000032G                     MOV     #$BFR+D$GVBL,R2         ; Point to destination name                      ;[TD001]
000510  012700  000000G                     MOV     #DSTNAM,R0              ; Point to specified destination name            ;[TD001]
000514  122022                   20$:       CMPB    (R0)+,(R2)+             ; Is name unique?                                ;[TD001]
000516  001346                              BNE     10$                     ; BR IF YES                                      ;[TD001]
000520  005301                              DEC     R1                      ; MORE TO CHECK?                                 ;[TD001]
000522  003374                              BGT     20$                     ; BR IF YES                                      ;**-8
000524  000261                              SEC                             ; ELSE SET C-BIT
000526                           30$:       RESMAP                          ; RESTORE PREVIOUS MAPPING
000532                                      RETURN                                                                           ;[TD001]
```

```
VFPDST - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:35  Page 23-2
Symbol table

Work  file writes: 0
Size of work file: 14348 Words  ( 57 Pages)
Size of core pool: 15496 Words  ( 59 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:45.18
SY:VFPDST.V2,[132,134]VFPDST/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPDST
```

```
                                  .SBTTL  TPARS STATE TABLES
                        ;
                        ; TPARS STATE TABLES
                        ;
000052                            ISTAT$  DTEST,DTEKW
                        ;
                        ; DTE$DF
                        ;
000052                            STATE$  DTEDF
000052                             TRAN$   %DTE$DF%,,,1,SYNERR

000052                            STATE$
000052                             TRAN$    $DIGIT,,DTEAST          ; START OF DTE ADDRESS

000052                            STATE$  DTEAD
000052                             TRAN$   $DIGIT,DTEAD,DTEDIG     ; DTE ADDRESS
000052                             TRAN$   <',>,,DTEAEN            ; END OF DTE ADDRESS

000052                            STATE$
000052                             TRAN$    !LINID                  ; LINE-ID

000052                            STATE$
000052                             TRAN$    <',>

000052                            STATE$
000052                             TRAN$    $NUMBR,,HSHTBL          ; HASH TABLE SIZE

000052                            STATE$
000052                             TRAN$   <',>

000052                            STATE$
000052                             TRAN$    $NUMBR,,CNTTIM          ; COUNTER TIMER VALUE

000052                            STATE$
000052                             TRAN$   !END,$EXIT,DTEEND
000052                             TRAN$   <',>

000052                            STATE$
000052                             TRAN$   "ON",DTENW,,DE.ON,DTEFLG       ; STATE
000052                             TRAN$   "OFF",DTENW,,DE.OFF,DTEFLG

000052                            STATE$  DTENW
000052                             TRAN$   !END,$EXIT,DTEEND
000052                             TRAN$   <',>

000052                            STATE$
000052                             TRAN$   $STRNG,,DTENET          ; NETWORK NAME PARAMETER

000052                            STATE$  END
000052                             TRAN$    $EOS,$EXIT
00005                              TRAN$    <';>,$EXIT
```

```
VFPDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:37      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

LF.SER   = 000040           #5-69
LF.TIM   = 000010           #5-69
LF.UNL   = 020000           #5-69
LF.X2P   = 000000           #5-69
LINCTL   = ******   GX       17-509
LINNAM   = ******   GX       17-496
LINUNT   = ******   GX       17-511
LNKEND   = ******   GX       15-420
LN$OFF   = 000001            15-402
LN$ON    = 000000            15-405
LN.CLO   = 000000           #5-69
LN.DUM   = 000005           #5-69
LN.LOA   = 000004           #5-69
LN.LOO   = 000003           #5-69
LN.OAU   = 000003           #5-69
LN.OFF   = 000001           #5-69
LN.ON    = 000000           #5-69
LN.OOP   = 000004           #5-69
LN.OPE   = 000001           #5-69
LN.REF   = 000002           #5-69
LN.SER   = 000002           #5-69
LN.STA   = 000017           #5-69
LN.SUB   = 000360           #5-69
LN.TRI   = 000006           #5-69
L$CHTB     000030           *15-414
L$CTEN     000032           *15-415
L$CTIM     000040           *15-400
L$DTEA     000020            15-379      16-460
L$DTEL     000017           *15-385
L$GLEN     000134            15-364      15-411
L$LEN      000122            15-361      15-364      15-408      15-411      16-457
L$LLCH     000016           *15-399
L$NETW     000114            15-388
L$NMST     000003           *15-402     *15-405      15-406
L$OMST     000002           *15-406
L$SLN      000004           *15-395
L$TCLZ     000044           *15-401
L.COST     000015           #5-69
L.CTL      000012           #5-69       17-509
L.CVA      177776           #5-69
L.DDM      000002           #5-69       17-507
L.DDS      000004           #5-69
L.DLC      000003           #5-69
L.DLM      000006           #5-69
L.DLS      000010           #5-69
L.FLG      000000           #5-69       17-516
L.KRBA     000016           #5-69
L.LEN    = 000022           #5-69
L.MPF      000022           #5-69
L.NMST     000020           #5-69
L.NSTA     000014           #5-69
L.OWNR     000021           #5-69
```

```
                                              .SBTTL  PSN$DF ACTION ROUTINES
                                              .ENABL  LSB                                                         ;[TD001]


                                     ;
                                     ; NETWORK NAME
                                     ;
000052                               PSNNM:   VNPDBG  VFPPSN              ; VNP debugging break point            ;[TD001]
000052  005767  000000G                       TST     $PSIPT              ; Have we already found a PSN$DF?      ;[TD001]
000056  001121                                BNE     101$                ; Branch if yes - error                ;[TD001]
000060  012700  000006'                       MOV     #PSNNML,R0          ; Clear net. name length               ;[TD001]
000064  105020                                CLRB    (R0)+                                                      ;[TD001]
000066  012701  000020                        MOV     #MAXID,R1           ; Clear name to spaces                 ;[TD001]
000072                                        CALL    SPFILL                                                     ;[TD001]
000076  026727  000000G 000020                CMP     .PSTCN,#MAXID       ; Check if name too long               ;[TD001]
000104  101111                                BHI     102$                                                       ;[TD001]
000106  016701  000000G                       MOV     .PSTCN,R1           ; If not too long, get actual length   ;[TD001]
000112  012702  000006'                       MOV     #PSNNML,R2                                                 ;[TD001]
000116  110122                                MOVB    R1,(R2)+            ; Insert actual length                 ;[TD001]
000120  016700  000000G                       MOV     .PSTPT,R0           ; Source of network name               ;[TD001]
000124                                        CALL    MOVE                ; Move network name to buffer          ;[TD001]
000130  005067  000000'                       CLR     PSNFLG              ; Initialize flags word                ;[TD001]
000134  005067  000002'                       CLR     PSNLO               ; Clear low subaddress value           ;[TD001]
000140  005067  000004'                       CLR     PSNHI               ; Clear high subaddress value          ;[TD001]
000144                                        RETURN                                                             ;[TD001]
                                                                                                                 ;**-25
                                     ;
                                     ; NUMBER OF PORTS
                                     ;
000146  005767  000000G              PORT:    TST     .PNUMH              ; IS THIS A SINGLE WORD VALUE?
000152  001071                                BNE     103$                ; BR IF NO - ERROR                     ;[TD001]
000154  016700  000000G                       MOV     .PNUMB,R0           ; GET PORT NUMBER                      ;[TD001]
000160  003466                                BLE     103$                ; BR IF ILLEGAL PORT NUMBER            ;[TD001]
000162  020027  000400                        CMP     R0,#PORTMX          ; IS THIS A LEGAL PORT NUMBER?         ;[TD001]
000166  101063                                BHI     103$                ; BR IF NO - ERROR                     ;[TD001]
000170  010067  000000G                       MOV     R0,PORTNO           ; SAVE PORT NUMBER                     ;**-5
000174                                        RETURN
                                     ;
                                     ; FLAGS WORD                                                                ;[TD001]
                                     ;                                                                           ;[TD001]
000176  005767  000000G              PSNFG:   TST     .PNUMH              ; Is this a double word value?         ;[TD001]
000202  001060                                BNE     104$                ; Branch if yes - error                ;[TD001]
000204  016767  000000G 000000'               MOV     .PNUMB,PSNFLG       ; Save flags word                      ;[TD001]
000212  005067  000000G                       CLR     $PFLAG              ; Initialise PSI characteristics word  ;[TD001]
000216  032767  000004  000000'               BIT     #HF$HOS,PSNFLG      ; Check for host flag                  ;[TD001]
000224  001403                                BEQ     20$                                                        ;[TD001]
000226  052767  000000G 000000G               BIS     #PF.XGA,$PFLAG      ; If host, set XGA-present flag        ;[TD001]
000234  032767  000010  000000'  20$:         BIT     #HF$GWY,PSNFLG      ; Check for gateway flag               ;[TD001]
000242  001407                                BEQ     30$                                                        ;[TD001]
000244  052767  000000G 000000G               BIS     #PF.PSV,$PFLAG      ; If gateway, set PSV-present flag     ;[TD001]
000252  032767  000020  000000'               BIT     #HF$XDF,PSNFLG      ; If gateway, MUST have extended data  ;[TD001]
000260  001431                                BEQ     104$                                                       ;[TD001]
000262  032767  000020  000000'  30$:         BIT     #HF$XDF,PSNFLG      ; Check for extended data flag         ;[TD001]
000270  001403                                BEQ     40$                                                        ;[TD001]
000272  052767  000000G 000000G               BIS     #PF.XDF,$PFLAG      ; If extended data, indicate in $PFLAG ;[TD001]
000300  032767  000000G 000000G  40$:         BIT     #PF.XDF,$PFLAG      ; Check for extended data format       ;[TD001]
000306  001004                                BNE     50$                                                        ;[TD001]
000310  126727  000006' 000006                CMPB    PSNNML,#PNAMMX      ; If not extended, check network name  ;[TD001]
```

Table of contents

```
     5-   56    MACRO DEFINITIONS
     6-   94    LOCAL SYMBOL DEFINITIONS
     7-  103    LOCAL DATA
     8-  143    ERROR MESSAGES
     9-  154    LOOK FOR PVC$DF MACRO
    10-  186    TPARS STATE TABLES
    11-  276    PVC$DF ACTION ROUTINES
    14-  412    ALLXCB - ALLOCATE X25 CIRCUIT BLOCK
    15-  461    CKPVC0 - Check for unique PVC name [new format]                      ;[TD001]
    16-  499    CKPVC1 - Check for unique PVC name [old format]                      ;[TD001]
    17-  535    ALPVC0 - Allocate PVC name block [new format] & circuit block        ;[TD001]
    18-  587    ALPVC1 - Allocate PVC name block [old format] & circuit block        ;[TD001]
    19-  635    FNDPRT - FIND FREE PORT ENTRY                                        ;[TD001]
    20-  678    INIXCB - INITIALIZE X25 CIRCUIT BLOCK
    21-  726    FNDHSH - FIND HASH TABLE ENTRY
    22-  772    BIASX - BIAS XPOOL ADDRESS FOR APR6
    23-  792    PVDLM - CHECK FOR DLM PVC
```

```
                                                                        G 1
VFGSVC - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 02:30  Page 17-2
Symbol table

SF.SVC= 000002           S$LFT    000012        T$LTCL   000024        Z.NAM = ****** GX       $NUMBR= 000002
SF.UNL= 000040           S$LNK    000000        T$LTIM   000026        $ALPHA= 000022          $PDVNM= ****** GX
SLTA    000016R      002 S$OWNR   000030        T$LTPR   000014        $ALPOL= ****** GX       $PDVTA= ****** GX
STASVC  000014R      002 S$PKSZ   000004        T$LTPS   000020        $ANY  = 000020          $PSIPT= ****** GX
STRNXT  001646R          S$RCL    000013        T$NAPL   000004        $BFR  = ****** GX       $PUTAD= ****** GX
SVCDF   000000R      003 S$STA    000003        T$NFE    000000        $BLANK= 000006          $QSVC   000000RG
SVCDTE  000110R      003 S$TIM    000014        T$NLEN   000010        $CAT5 = ****** GX       $RADDR= ****** GX
SVCFLG  000054R      003 S$TMR    000016        T$NNUL   000002        $CDTB = ****** GX       $RAD50= 000016
SVCKW   000000RG     004 S$WND    000006        T$NOFL   000006        $CEACC= ****** GX       $RBLCK= ****** GX
SVCST   000000RG     003 S$$WRG=  000000        T$NRNI   000042        $DIGIT= 000024          $SAVAL= ****** GX
SVCTL   000114R          S$$YSZ=  007600        T$NRPL   000005        $DNUMB= 0000            $SAVRG= ****** GX
SVCWND  000042R      003 S.COST   000001        T$NRUL   000007        $EOS  = 000012          $SLTA = ****** GX
SVDTED  000476R          S.FLG    000000        T$NVR    000001        $ERR1T= ****** GX       $SLTMA= ****** GX
SVDTEE  000522R          S.LEN    000004        T$RPRI   000040        $ERR2I= ****** GX       $STRNG= 000004
SVDTES  000440RG         S.NMST   000002        T$SVC    000034        $ERR3D  000056R     002 $SUBXP= 000010
SVFLAG  000316R          S.OWNR   000003        T$T5     000030        $ERR3E  000102R     002 $TRIB = ****** GX
SVNAM   000052R          T$FLAG   000044        T$T6     000032        $ERR3F  000126R     002 $XLINK= ****** GX
SVOWN   000122R      003 T$LIF    000013        T$$KMG=  000000        $ERR3G  000152R     002 $$HTIM= ****** GX
SVOWNR  000550R          T$LIFL   000013        T$$MIN=  000000        $ERR3H  000202R     002 $$LTIM= ****** GX
SVPKT   000236R          T$LIFO   000013        VF$RCV=  100000        $ERR3I  000230R     002 $$$FLG= 177777
SVRCLL  000342R          T$LIFS   000013        VF$XMT=  040000        $ERR3J  000254R     002 $$$KEY= 000000
SVRRTM  000374R          T$LIN    000000        V$FLG    000000        $ERR3K  000272R     002 $$$R  = 000010
SVSTA   000164R          T$LIPS   000006        V$LEN    000022        $ERR3M  000344R     002 $$$STA= 000000
SVWND   000262R          T$LLD    000012        V$RCV    000002        $ERR3Z  000400R     002 $$$TMP= 000000R      005
SYNERR= ****** GX        T$LLDC   000045        V$XMT    000012        $EXIT = 000000          .PCHAR= ****** GX
SYSFDB= ****** GX        T$LLDL   000012        V$$CTR=  001000        $FAIL = 177777          .PNUMB= ****** GX
S$ADSZ  000007           T$LLDO   000012        WNDSVC   000022R   002 $GETAD= ****** GX       .PNUMH= ****** GX
S$CTL   000002           T$LLDS   000012        X$$DBT=  000000        $GETRV= ****** GX       .PSTCN= ****** GX
S$DTE   000020           T$LLEN   000046        Z.DAT  = ****** GX     $LAMDA= 000000          .PSTPT= ****** GX
S$FLG   000010           T$LOPR   000002        Z.MAP  = ****** GX     $LLCTA= ****** GX       .TPARS= ****** GX
S$LEN   000032

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001702    001   (RW,I,LCL,REL,CON)
DATA    000430    002   (RW,D,LCL,REL,CON)
$STATE  000130    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 54
Work  file writes: 63
Size of work file: 20236 Words  ( 80 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:55.10
SY:VFGSVC.V2,[132,134]VFGSVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFGSVC
```

```
                                                     H 1
VFGSVC       CREATED BY  MACRO  ON 29-JUN-85 AT 02:31       PAGE 1
```

```
     92                                              .SBTTL  ERROR MESSAGES
     93
     94                                      ;
     95                                      ; ERROR MESSAGES
     96                                      ;
     97                                              .ENABL  LC
     98 000004                                       NTLER$  ,2J,8B,CERR,RTSPC,CFLIN,<maximum retransmit count>
     99 000046                                       NTLER$  ,2K,8B,CERR,RTSPC,CFLIN,<retransmit timer value>
    100 000106                                       NTLER$  ,2L,8B,CERR,RTSPC,CFLIN,<holdback timer value>
    101                                              .DSABL  LC
```

```
**FILE**ID**VFGX3P
```

```
VV      VV    FFFFFFFFFF     GGGGGGGG   XX      XX      333333     PPPPPPPP
VV      VV    FFFFFFFFFF     GGGGGGGG   XX      XX      333333     PPPPPPPP
VV      VV    FF           GG           XX      XX    33      33   PP      PP
VV      VV    FF           GG           XX      XX    33      33   PP      PP
VV      VV    FF           GG             XX  XX              33   PP      PP
VV      VV    FF           GG             XX  XX              33   PP      PP
VV      VV    FFFFFFFF     GG               XX              33     PPPPPPPP
VV      VV    FFFFFFFF     GG               XX              33     PPPPPPPP
VV      VV    FF           GG  GGGGGG     XX  XX              33   PP
VV      VV    FF           GG  GGGGGG     XX  XX              33   PP
  VV  VV      FF           GG      GG   XX      XX    33      33   PP                ....
  VV  VV      FF           GG      GG   XX      XX    33      33   PP                ....
    VV        FF             GGGGGG     XX      XX      333333     PP                ....
    VV        FF             GGGGGG     XX      XX      333333     PP                ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
VFGX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32      PAGE 2
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE            REFERENCES

$QX3P     000000 RG      #9-113
$RAD50  = 000016         #10-133
$RONLY  = ******          10-133       10-133     10-133
$STRNG  = 000004         #10-133
$SUBXP  = 000010         #10-133
$X3CAL  = ******   GX     11-219
$X3DBS  = ******   GX    *11-183       11-191
$X3DWS  = ******   GX    *11-202       11-210
$X3MBS  = ******   GX    *11-193
$X3MWS  = ******   GX    *11-212
$$$FLG  = 177777         #10-133
$$$KEY  = 177777         #10-133
.PNUMB  = ******   GX     11-229       11-231
.PNUMH  = ******   GX     11-227
.TPARS  = ******   GX     9-119
```

```
VFIND       CREATED BY  MACRO  ON 29-JUN-85 AT 02:33     PAGE 1
SYMBOL  CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE               REFERENCES

A.ACC      000020           #5-65
A.LNK      000000           #5-65
A.NAM      000002           #5-65      8-147      8-149      8-151
A.PUD      000010           #5-65
A.REM      000012           #5-65
A.UCB      000010           #5-65      8-153
CURUNM     000000 R         #6-74     *7-93       7-94      *7-95       7-98       7-104     *8-135      8-136     *8-137
                             8-140     8-155     *9-183      9-184     *9-185      9-188      9-192     *10-221    10-222
                            *10-223    10-226     10-235
D$ANN      000000            8-131
D$END    = 000104            8-128     9-176      10-214
D$RNN      000002            9-179     10-217
F.PSIZ   = ******    GX     *7-99     *8-128     *8-141     *9-176     *9-189     *10-214    *10-227
F.URBD   = ******    GX     *8-127    *8-129     *9-175     *9-177     *10-213    *10-215
LD.ALI   = 000030           #5-66
LD.CEX   = 000024           #5-66
LD.CIR   = 000035           #5-66
LD.LIN   = 000025           #5-66
LD.LOG   = 000031           #5-66
LD.MOD   = 000037           #5-66
LD.NOD   = 000027           #5-66
LD.OBJ   = 000032           #5-66
LD.PRO   = 000026           #5-66
LO.ADD     000022           #5-66
LO.ADR     000052           #5-66
LO.CLS     000002           #5-66
LO.CON     000006           #5-66
LO.COP     000010           #5-66
LO.COS     000020           #5-66
LO.CSR     000010           #5-66      5-66
LO.CTL     000032           #5-66
LO.DES     000002           #5-66      5-66
LO.DTE     000047           #5-66
LO.EVT     000004           #5-66
LO.GRO     000024           #5-66
LO.HOS     000040           #5-66
LO.HTM     000032           #5-66
LO.INA     000062           #5-66
LO.INC     000060           #5-66
LO.LDT     000046           #5-66
LO.LID     000030           #5-66
LO.LIN     000004           #5-66
LO.LTM     000034           #5-66
LO.MAC     000022           #5-66
LO.MDE     000024           #5-66
LO.NAM     000044           #5-66
LO.NET   = 000002           #5-66
LO.NNM     000022           #5-66
LO.OFL     000012           #5-66
LO.ONM     000004           #5-66
LO.OUT     000056           #5-66
LO.OWN     000026           #5-66
```

```
    120                                        .SBTTL  LOOK FOR CUG$DF MACRO
    121
    122                                ;+
    123                                ; $QCUG - LOOK FOR CUG$DF MACRO
    124                                ;
    125                                ; INPUTS:
    126                                ;       NONE
    127                                ;
    128                                ; OUTPUTS:
    129                                ;       C-BIT=SUCCESS/FAILURE
    130                                ;       R3,R4,R5=DESTROYED
    131                                ;-
    132                                                                                                   ;[TD001]
    133 000000                         $QCUG:: VNPDBG  VFPCUG          ; VNP debugging break point       ;[TD001]
    134 000000  012705  000000'                MOV     #CUGDF,R5       ; STATE TABLE ADDRESS             ;[TD001]
    135 000004  005001                         CLR     R1              ; FULL KEYWORD MATCH LENGTH       ;**-1
    136 000006  012702  000000'                MOV     #CUGKW,R2       ; KEYWORD TABLE ADDRESS
    137 000012  016703  000000G                MOV     CFGSZ,R3        ; RECORD LENGTH
    138 000016  012704  000000G                MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
    139 000022  005067  000000G                CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
    140 000026                                 CALL    .TPARS          ; GO DO THE PARSE
    141 000032  103003                         BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
    142 000034  005367  000000G                DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
    143 000040  001401                         BEQ     101$            ; IF EQ, YES
    144 000042                         20$:    RETURN
    145
    146                                ;
    147                                ; ERROR CONDITION
    148                                ;
    149 000044                         101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
VFPCUG     CREATED BY  MACRO  ON 29-JUN-85 AT 02:33      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

STRNXT  = ******  GX        11-215     11-223
SYNERR  = ******  GX        *9-139     *9-142
SYSFDB  = ******  GX       *14-297    *14-303    *15-337    *15-343    *18-423
S$$BAS  = ******             8-111      8-111      8-112      8-112      8-113      8-113      8-114      8-114
$ALPHA  = 000022           #10-153
$ALPOL  = ******  GX        18-419
$ANY    = 000020           #10-153
$BFR    = ******  GX        14-299     14-304     14-307     14-310     15-339     15-344     15-348     18-424     18-430
                           *18-434    *18-435     18-440    *18-445
$BLANK  = 000006           #10-153
$CEACC  = ******  GX        14-302     15-342     18-422
$DBG$   = ******             9-133
$DIGIT  = 000024           #10-153
$DNUMB  = 000014           #10-153
$EOS    = 000012           #10-153
$ERRM1    000126 R          #8-114     13-276
$ERRN2    000034 R          #8-111     11-203     13-275
$ERRN3    000056 R          #8-112     11-228
$ERRP4    000102 R          #8-113     13-274
$ERR1T  = ******  GX         9-149
$EXIT   = 000000           #10-153
$FAIL   = 177777           #10-153
$GETAD  = ******  GX        14-303     15-343     18-423
$GETRV  = ******  GX        14-297     15-337
$GPRM   = ******            10-153
$LAMDA  = 000000           #10-153
$NUMBR  = 000002           #10-153
$PFLAG  = ******  GX        13-252
$PSIPT  = ******  GX        14-295     15-335     18-448
$PUTAD  = ******  GX        18-447
$QCUG     000000 RG         #9-133
$RADDR  = ******  GX       *14-297    *14-303    *15-337    *15-343    *18-423
$RAD50  = 000016           #10-153
$RBLCK  = ******  GX        14-298    *14-316     15-338    *15-354     18-415    *18-452
$RONLY  = ******            10-153     10-153     10-153
$SAVAL  = ******  GX        14-294     15-334     18-414
$SAVRG  = ******  GX        12-233
$STRNG  = 000004           #10-153
$SUBXP  = 000010           #10-153
$$$FLG  = 177777           #10-153
$$$KEY  = 177777           #10-153
.PNUMB  = ******  GX        13-248     13-250
.PNUMH  = ******  GX        13-246
.TPARS  = ******  GX         9-140
```

```

                                        ;
                                        ; CLOSED USER GROUP NAME
                                        ;
000306                                  DACUNM:                                                                  ;[TD001]
000306  012700  000000G                         MOV     #DSACGL,R0              ; address of buffer to contain name      ;[TD001]
000312  012701  000006                          MOV     #CUNMMX,R1              ; max length if old-style system         ;[TD001]
000316                                          CALL    STORID                  ; store name in buffer                   ;[TD001]
000322  103401                                  BCS     101$                    ; carry set if name too long             ;[TD001]
000324                                          RETURN                                                                   ;**-8
                                        ;
                                        ; ERRORS
                                        ;
000326                                  101$:   EMSG$R  N2                      ; ILLEGAL CUG NAME
```

```
                                        .SBTTL  Look for DSN$DF macro

                                ;+
                                ; *** $QDSN - Look for DSN$DF macro
                                ;
                                ; INPUTS:
                                ;       None.
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=success/failure
                                ;       R3,R4,R5 destroyed
                                ;-

000000                          $QDSN:: VNPDBG  VFPDSN          ; VNP debugging break point
000000  012705  000000'                 MOV     #DSNDF,R5       ; State table address
000004  012701  000001          10$:    MOV     #1,R1           ; Full keyword match length, blanks significant
000010  012702  000000'                 MOV     #DSNKW,R2       ; Keyword table address
000014  016703  000000G                 MOV     CFGSZ,R3        ; Record length
000020  012704  000000G                 MOV     #CFGBF,R4       ; Record buffer address
000024  005067  000000G                 CLR     SYNERR          ; Clear syntax error flag
000030                                  CALL    .TPARS          ; Go do the parse
000034  103003                          BCC     20$             ; If CC, found what we were looking for
000036  005367  000000G                 DEC     SYNERR          ; Did syntax error occur?
000042  001401                          BEQ     101$            ; If EQ, yes
000044                          20$:    RETURN
                                ;
                                ; Error condition
                                ;
000046                          101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
**FILE**ID**VFPDST




VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD       SSSSSSSS   TTTTTTTTTT
VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD       SSSSSSSS   TTTTTTTTTT
VV      VV   FF           PP      PP   DD      DD   SS               TT
VV      VV   FF           PP      PP   DD      DD   SS               TT
VV      VV   FF           PP      PP   DD      DD   SS               TT
VV      VV   FF           PP      PP   DD      DD   SS               TT
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD     SSSSSS         TT
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD     SSSSSS         TT
VV      VV   FF           PP           DD      DD           SS       TT
VV      VV   FF           PP           DD      DD           SS       TT
  VV  VV     FF           PP           DD      DD           SS       TT           ....
  VV  VV     FF           PP           DD      DD           SS       TT           ....
    VV       FF           PP           DDDDDDDD     SSSSSSSS         TT           ....
    VV       FF           PP           DDDDDDDD     SSSSSSSS         TT           ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        .SBTTL  CKDSN1 - Check destination name is unique [old format]          ;[TD001]
                                ;+                                                                              ;[TD001]
                                ;                                                                               ;[TD001]
                                ; *** CKDSN1 - Check destination name is unique [old format]                    ;[TD001]
                                ;                                                                               ;[TD001]
                                ; INPUTS:                                                                       ;[TD001]
                                ;       R0 - ADDRESS OF DESTINATION DESCRIPTOR LISTHEAD                         ;[TD001]
                                ;                                                                               ;[TD001]
                                ; OUTPUTS:                                                                      ;[TD001]
                                ;       CARRY CLEAR - DESTINATION NAME IS UNIQUE                                ;[TD001]
                                ;       CARRY SET   - DESTINATION NAME IS NOT UNIQUE                            ;[TD001]
                                ;       Registers R3-R5 preserved                                               ;[TD001]
                                ;                                                                               ;[TD001]
                                ;-                                                                              ;[TD001]
000534                          CKDSN1: CALL    $SAVAL                  ; SAVE ALL REGISTERS                    ;[TD001]
000540                                  SAVMAP                          ; SAVE CURRENT MAPPING                  ;[TD001]
000544                                  GETRV   R0,#2                   ; READ IN ADDRESS LISTHEAD              ;[TD001]
000562  016700  000000G         10$:    MOV     $BFR,R0                 ; GET ADDRESS OF NEXT BLOCK             ;[TD001]
000566  000241                          CLC                             ; ASSUME SUCCESS                        ;[TD001]
000570  001426                          BEQ     30$                     ; BR IF END OF LIST                     ;[TD001]
000572                                  CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS             ;[TD001]
000602                                  GETAD   R0,#D$FLEN              ; Read in destination name block        ;[TD001]
000620  012702  000004G                 MOV     #$BFR+D$NAM,R2          ; Point to destination name             ;[TD001]
000624  012700  000000G                 MOV     #DSTNAM,R0              ; Point to specified destination name   ;[TD001]
000630  012701  000006                  MOV     #DSNAMX,R1              ; Length of field                       ;[TD001]
000634  122022                  20$:    CMPB    (R0)+,(R2)+             ; Is name unique?                       ;[TD001]
000636  001351                          BNE     10$                     ; BR IF YES                             ;[TD001]
000640  005301                          DEC     R1                      ; MORE TO CHECK?                        ;[TD001]
000642  003374                          BGT     20$                     ; BR IF YES                             ;[TD001]
000644  000261                          SEC                             ; ELSE SET C-BIT                        ;[TD001]
000646                          30$:    RESMAP                          ; RESTORE PREVIOUS MAPPING              ;[TD001]
000652                                  RETURN                                                                  ;[TD001]
```

```
VFPDST     CREATED BY  MACRO  ON 29-JUN-85 AT 02:36     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE           REFERENCES

ALDST0     000654 R      13-273   #16-374
ALDST1     001152 R      13-281   #17-435
BCDLEN     001642 R      16-380    17-439   #20-566
CERR     = ******   GX   7-95      7-96      7-97      7-98      7-99      7-100     7-101
CFGBF    = ******   GX   8-126
CFGSZ    = ******   GX   8-125
CFLIN    = ******   GX   7-95      7-96      7-97      7-98      7-99      7-100     7-101
CKDSN0     000412 R      13-271   #14-307
CKDSN1     000534 R      13-279   #15-342
CUNMMX   = 000006       #6-87      17-469
DSACGL   = ******   GX  *10-195    17-471
DSACUG   = ******   GX   17-473
DSARCT   = ******   GX  *10-205    16-409    17-478
DSASHI   = ******   GX  *10-202    16-393    17-452
DSASLO   = ******   GX  *10-201    16-392    17-451
DSCMCT   = ******   GX  *10-203    13-258    16-410    17-479
DSCVCT   = ******   GX  *10-204    13-258
DSFLG    = ******   GX  *8-123     13-261    18-507
DSNAM      000056 R     #10-191
DSNAMX   = 000006       #6-85      10-192    15-352    17-454
DSOBJ      000170 R     #11-222
DSPRI      000162 R     #11-217
DSTDF      000000 R      8-121
DSTEND     000266 RG    #13-257
DSTINS     001504 R      16-417    17-486   #18-505
DSTKMX   = 000006       #6-86      12-243
DSTKW      000000 RG     8-124    #9-143
DSTNAM   = ******   GX   14-319    15-351    17-458
DSTNML   = ******   GX   10-191    14-316    17-456
DSTOBJ   = ******   GX   11-222    11-232    16-391    17-450
DSTPRI   = ******   GX   11-217    16-390    17-449    18-517
DSTSK      000240 R     #12-242
DSTST      000000 RG    #9-143
DSTTKL   = ******   GX  *10-196    12-242    17-461
DSTTSK   = ******   GX   17-462
DSTVR0   = ******   GX   16-377    16-401    16-406
DSTVR1   = ******   GX   16-379    16-404    17-438    17-476
DS.X29   = ******   GX   13-261    18-507
D$CUGN     000022        17-468    17-472
D$DATL     000030       *16-413   *17-482
D$DSL      000022        14-315    16-407
D$DST      000010        17-453    17-457
D$DTEL     000031       *16-409   *17-478
D$FLEN     000032        15-349    17-437
D$GLEN     000032        16-376
D$GVBL     000032        14-314    14-318    16-402
D$NAM      000004        15-350    17-460
D$NOD      000010        16-394    16-398
D$OBJ      000003       *16-391   *17-450
D$PRI      000002       *16-390   *17-449    18-516    18-517
D$SHI      000020       *16-393   *17-452
D$SLO      000016       *16-392   *17-451
```

```
                                        .SBTTL  LINE-ID SUBEXPRESSION

                                ;
                                ; LINE-ID SUBEXPRESSION
                                ;
000052                                  STATE$  LINID
000052                                   TRAN$   $RAD50,,DEVNAM          ; DEVICE NAME

000052                                  STATE$
000052                                   TRAN$   <'->

000052                                  STATE$
000052                                   TRAN$   $NUMBR,,DEVCTL          ; CONTROLLER NUMBER

000052                                  STATE$
000052                                   TRAN$   <'->,LINID1
000052                                   TRAN$   $LAMDA,$EXIT

000052                                  STATE$  LINID1
000052                                   TRAN$   $NUMBR,$EXIT,DEVUNT     ; UNIT NUMBER

000052                                  STATE$
```

```
VFPDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:37      PAGE 3
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE                 REFERENCES

L.UNT     000013        #5-69       17-511
NEXT    = ******  GX    *12-252     12-259
NUMDTE  = 000020        #6-89       15-352
N$$VCT  = ******         15-372     16-456
PCKBCD  = ******  GX     12-275
PF.PSV  = ******  GX     15-362     15-409
PF.XDF  = ******  GX     12-280
RTSPC   = ******  GX     8-125      8-126      8-127      8-128      8-129      8-130
R$R0    = 000002        #6-94
R$R1    = 000004        #6-95       15-416
R$R2    = 000006        #6-96
R$R3    = 000010        #6-97
R$R4    = 000012        #6-98      *16-468    *17-520
R$R5    = 000014        #6-99
SF.ACT  = 000200        #5-69
SF.ENA  = 000100        #5-69
SF.LPB  = 000004        #5-69
SF.MFL  = 000040        #5-69
SF.PAC  = 000020        #5-69
SF.REA  = 000010        #5-69
SF.SER  = 000001        #5-69
SF.SVC  = 000002        #5-69
SF.UNL  = 000040        #5-69
SPACE   = 000040        #6-86
SPFILL  = ******  GX     12-279
STRNXT  = ******  GX     12-253     12-261
SYNERR  = ******  GX    *9-154     *9-157
SYSFDB  = ******  GX    *12-284    *15-416    *16-449    *16-457
S$$BAS  = ******         8-125      8-125      8-126      8-126      8-127      8-127      8-128      8-128      8-129
                         8-129      8-130      8-130

S.COST    000001        #5-69
S.FLG     000000        #5-69
S.LEN     000004        #5-69
S.NMST    000002        #5-69
S.OWNR    000003        #5-69
$ALPHA  = 000022        #10-170
$ALPOL  = ******  GX     15-367
$ANY    = 000020        #10-170
$BFR    = ******  GX     12-285     15-373     15-378    *15-385     15-387    *15-395    *15-399    *15-400    *15-401
                        *15-402    *15-405     15-406    *15-406    *15-414    *15-415     16-452     16-459

$BLANK  = 000006        #10-170
$CEACC  = ******  GX     15-372     16-456
$DBG$   = ******         9-148
$DIGIT  = 000024        #10-170
$DNUMB  = 000014        #10-170
$EOS    = 000012        #10-170
$ERRO0    000154  R     #8-128      14-344
$ERROP    000214  R     #8-129      15-428
$ERROQ    000266  R     #8-130      15-429
$ERRO3    000036  R     #8-125      12-266     15-426
$ERRO4    000070  R     #8-126      13-312
$ERRO5  = ******  GX     13-324
```

```
    000316  101004                  BHI     102$                    ; Error if name too long                ;[TD001]
    000320                  50$:    RETURN                                                                  ;[TD001]
                            ;                                                                               ;[TD001]
                            ; ERROR MESSAGES                                                                ;[TD001]
                            ;                                                                               ;[TD001]
    000322                  101$:   EMSG$R  P1                      ; ERROR - ILLEGAL OCCURENCE OF PSN$DF   ;[TD001]
    000330                  102$:   EMSG$R  P2                      ; ERROR - ILLEGAL NETWORK NAME          ;[TD001]
    000336                  103$:   EMSG$R  P3                      ; ERROR - ILLEGAL NUMBER OF PORTS       ;[TD001]
    000344                  104$:   EMSG$R  P4                      ; ERROR - ILLEGAL FLAGS WORD            ;[TD001]
                                                                                                            ;[TD001]
                                    .DSABL  LSB                                                             ;[TD001]
```

```
                .TITLE  VFPPVC - INITIAL CONFIG FILE SCAN ACTION ROUTINES
                .IDENT  /V05.00/
                .ENABL  LC                                                          ;[TD001]
                                                                                    ;**-1
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-82
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                                                  H  1
VFGSVC     CREATED BY  MACRO  ON 29-JUN-85 AT 02:31     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE          REFERENCES

CFERR   = ******  GX    7-140     7-141     7-142     7-143     7-144     7-145     7-146     7-147     7-148
                        7-149
CFGBF   = ******  GX    8-171
CFGSZ   = ******  GX    8-170
COUNT     000024 R     #6-129    *12-389    13-460    16-613   *17-653
CTLSVC    000021 R     #6-126    *10-281    13-452
CVTBUF    000054 R     #6-134    *17-648    17-649
DTEEND  = 000044 R     #6-132     12-397
DTELEN  = 000017       #6-130     6-131     6-133     12-384    12-410    13-461
DTEPCK    000044 R     #6-133     12-412    13-463
DTESVC    000025 R     #6-131     12-385    12-390    12-411
E$NBR     000014       #5-62
E$NBS     000020       #5-62
E$NCR     000034       #5-62
E$NCS     000036       #5-62
E$NIC     000044       #5-62
E$NLEN    000050       #5-62
E$NLLA    000012       #5-62
E$NLNK    000000       #5-62
E$NML     000040       #5-62
E$NMR     000024       #5-62
E$NMS     000030       #5-62
E$NNOD    000002       #5-62
E$NRT     000042       #5-62
E$NRTP    000005       #5-62
E$NSEG    000010       #5-62
E$NTIM    000046       #5-62
E$NUSE    000004       #5-62
E$STRT    000006       #5-62
FLGSVC    000010 R     #6-121    *11-338    12-374    13-455
FMT8    = ******  GX    7-147     7-148     7-149
FMT8B   = ******  GX    7-140     7-141     7-142     7-143     7-144     7-145     7-146
FM.8    = 000000       #7-147    #7-148    #7-149
FM.8B   = 000000       #7-140    #7-141    #7-142    #7-143    #7-144    #7-145    #7-146
FNDOWN    001214 R      13-427   #14-509
F.RSIZ  = ******  GX   *13-468   *13-470   *13-475   *15-568   *15-573
HTIMR     000002 R     #6-118    *13-432    15-590
H$SVC     000036        13-469
I$$AS   = ******        16-623
LF.ACT  = 100000       #5-65
LF.BRO  = 000400       #5-65
LF.BWT  = 000007       #5-65
LF.ENA  = 002000       #5-65
LF.LPB  = 001000       #5-65
LF.MDC  = 000100       #5-65
LF.MFL  = 004000       #5-65
LF.MTP  = 000020       #5-65
LF.PAC  = 000200       #5-65
LF.RDY  = 040000       #5-65
LF.REA  = 010000       #5-65
LF.SER  = 000040       #5-65
LF.TIM  = 000010       #5-65
```

```
                                                                  I  1
```

```
                                                 .SBTTL  $QX2P - LOOK FOR X2P$DF MACRO

                                        ;+
                                        ;
                                        ; $QX2P - LOOK FOR X2P$DF MACRO
                                        ;
                                        ; INPUTS:
                                        ;        NONE
                                        ;
                                        ; OUTPUTS:
                                        ;        ALL REGISTERS DESTROYED
                                        ;
                                        ;-
 000144  012705  000000'                $QX2P:: MOV     #X2PDF,R5       ; STATE TABLE ADDRESS
 000150  005001                                 CLR     R1              ; FULL KEYWORD MATCH LENGTH
 000152  012702  000000'                        MOV     #X2PKW,R2       ; KEYWORD TABLE ADDRESS
 000156  016703  000000G                        MOV     CFGSZ,R3        ; RECORD LENGTH
 000162  012704  000000G                        MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
 000166  005067  000000G                        CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
 000172                                         CALL    .TPARS          ; GO DO THE PARSE
 000176  103003                                 BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
 000200  005367  000000G                        DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
 000204  001401                                 BEQ     101$            ; IF EQ, YES
 000206                                 20$:    RETURN
                                        ;
                                        ; ERRORS
                                        ;
 000210                                 101$:   EMSG$R  1T              ; SYNTAX ERROR
```

Table of contents

```
     5-   55    MACRO DEFINITIONS
     6-   62    CONSTANT DEFINITIONS
     7-   72    LOCAL DATA
     8-   85    ERROR MESSAGES
     9-  100    $QX3P - LOOK FOR X3P$DF MACRO
    10-  129    X3P$DF STATE TABLE
    11-  176    X3P$DF ACTION ROUTINES
```

```
VFGX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:32     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

CALL             9-119      11-181     11-189     11-200     11-208
DBGTP$          #10-133    #10-169
EMSG$R          #5-60       9-127      11-185     11-195     11-196     11-204     11-214     11-215     11-237
ISTAT$          #5-60       10-133
MTRAN$          #10-133
NTLER$          #5-60       8-91       8-92       8-93       8-94       8-95
RETURN           9-123      11-184     11-194     11-203     11-213     11-233     11-236     11-244
STATE$          #5-60       10-137    #10-140    #10-143    #10-146    #10-149    #10-152    #10-155    #10-158    #10-161
                #10-164    #10-167    #10-171    #10-174
TRAN$           #5-60      #10-138    #10-141    #10-144    #10-147    #10-150    #10-153    #10-156    #10-159    #10-162
                #10-165    #10-168    #10-169    #10-172
```

```
VFIND       CREATED BY  MACRO  ON 29-JUN-85 AT 02:33      PAGE 2
SYMBOL  CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE              REFERENCES

LO.PAR     000010           #5-66
LO.PRI     000012           #5-66
LO.RET     000064           #5-66
LO.RPA     000030           #5-66
LO.SCR     000016           #5-66
LO.SEG     000054           #5-66
LO.SKA     000014           #5-66
LO.SKN     000014           #5-66
LO.SNM     000035           #5-66
LO.TPA     000020           #5-66
LO.TRB     000034           #5-66
LO.TRI     000030           #5-66
LO.UNT     000033           #5-66
LO.VCT     000014           #5-66
LO.VER     000042           #5-66
LR.ACC     000020           #5-66
LR.ACO     000052           #5-66
LR.ADD     000012           #5-66
LR.ALI     000002           #5-66
LR.BLK     000010           #5-66
LR.CTL     000004           #5-66
LR.DES     000010           #5-66
LR.HLP     000002           #5-66
LR.LIN     000002           #5-66        5-66      5-66      5-66      5-66      5-66      5-66
LR.NAM     000012           #5-66
LR.PAS     000041           #5-66
LR.PRO     000002           #5-66
LR.STS     000000           #5-66
LR.TRI     000006           #5-66
LR.TYP     000002           #5-66
LR.UCB     000016           #5-66        8-153
LR.UIC     000020           #5-66
LR.UNT     000005           #5-66
LS.ALI   = 000016           #5-66
LS.CEX   = 000001           #5-66
LS.CIR   = 000033           #5-66
LS.CXO   = 010000           #5-66
LS.CX5   = 000400           #5-66
LS.ECH   = 001000           #5-66
LS.FDX   = 004000           #5-66
LS.HDX   = 010000           #5-66
LS.HLP   = 000012           #5-66
LS.LIN   = 000003           #5-66
LS.LMC   = 000007           #5-66
LS.LOG   = 000020           #5-66
LS.MOD   = 000036           #5-66
LS.NOD   = 000014           #5-66
LS.NTI   = 000013           #5-66
LS.OBJ   = 000022           #5-66
LS.OPT   = 000400           #5-66
LS.PRO   = 000002           #5-66
LS.TOP   = 002000           #5-66
```

```
                                   .SBTTL  CUG$DF STATE TABLES

000052                             ISTAT$  CUGST,CUGKW
                           ;
                           ; CUG$DF STATE TABLES
                           ;
000052                             STATE$  CUGDF
000052                               TRAN$   %CUG$DF%,,,1,SYNERR

000052                             STATE$
000052                               TRAN$   $RAD50,,CUNAM          ; CLOSED USER GROUP NAME              ;[TD001]
                                                                                                         ;**-1
000052                             STATE$
000052                               TRAN$   <',>

000052                             STATE$
000052                               TRAN$   $DIGIT,,CUNBST         ; START OF CLOSED USER GROUP NUMBER

000052                             STATE$  CUNXT
000052                               TRAN$   $DIGIT,CUNXT,CUGDIG    ; ANOTHER DIGIT
000052                               TRAN$   $LAMDA,,CUNBEN         ; END OF CUG NUMBER

000052                             STATE$
000052                               TRAN$  !END,$EXIT,CUEND
000052                               TRAN$  <',>

000052                             STATE$
000052                               TRAN$   $NUMBR,,CUFG           ; CLOSED USER GROUP FLAGS BYTE
                           ;
                           ; CHECK FOR END OF SOURCE LINE
                           ;
000052                             STATE$  END
000052                               TRAN$   <';>,$EXIT
000052                               TRAN$   $EOS,$EXIT

000052                             STATE$
```

```
VFPCUG      CREATED BY  MACRO  ON 29-JUN-85 AT 02:33      PAGE 3

MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME      REFERENCES

CALL              9-140    11-196    11-215    11-223    12-239    13-257    13-259    13-265    13-267    14-294
                 14-297    14-302    14-303    15-334    15-337    15-342    15-343    16-377    17-395    18-414
                 18-419    18-422    18-423    18-432    18-441    18-447    18-450
CEACC$          #5-62      14-302    15-342    18-422
CUGDF$          #5-61      5-65
DBGTP$          #10-153   #10-171   #10-175   #10-184
EMSG$R          #5-61      9-149     11-203    11-228    13-274    13-275    13-276
GETAD           #5-62      14-303    15-343    18-423
GETRV           #5-62      14-297    15-337
ISTAT$          #5-61      10-153
MTRAN$          #10-153
NTLER$          #5-61      8-111     8-112     8-113     8-114
PHBDF$          #5-61      5-66
PUTAD           #5-61      18-447
RAND            #14-297    14-297   #14-303    14-303   #15-337    15-337   #15-343    15-343   #18-423    18-423
                #18-447    18-447
RESMAP          #5-75      14-316    15-354    18-452
RETURN            9-144    11-199    11-216    11-224    12-240    13-270    14-317    15-355    16-378    17-396
                 18-453
SAVMAP          #5-71      14-298    15-338    18-415
STATE$          #5-61      10-157   #10-160   #10-163   #10-166   #10-169   #10-173   #10-177   #10-182   #10-186
TRAN$           #5-61     #10-158   #10-161   #10-164   #10-167   #10-170   #10-171   #10-174   #10-175   #10-178
                #10-183   #10-184
VNPDBG          #5-63      9-133
```

```
                                     .SBTTL  DSC$DF ACTION ROUTINES


                           ;
                           ; STORE CALL DATA MASK
                           ;
000334  012700  000000G    DCMSK:  MOV     #DSCMSK,R0              ; POINT TO CALL DATA MASK STORAGE
000340  012702  000000G            MOV     #DSCMCT,R2              ; POINT TO CALL DATA MASK COUNT
000344                             CALL    STRHEX                  ; STORE THE HEX VALUE
000350  103401                     BCS     101$                    ; BR IF ILLEGAL VALUE
000352                             RETURN
                           ;
                           ; ERRORS
                           ;
000354                     101$:   EMSG$R  M3                      ; ILLEGAL CALL DATA MASK

                           ;
                           ; STORE CALL DATA VALUE
                           ;
000362  012700  000000G    DCVAL:  MOV     #DSCVAL,R0              ; POINT TO CALL DATA VALUE STORAGE
000366  012702  000000G            MOV     #DSCVCT,R2              ; POINT TO CALL DATA VALUE COUNT
000372                             CALL    STRHEX                  ; STORE THE HEX VALUE
000376  103401                     BCS     101$                    ; BR IF ILLEGAL VALUE
000400                             RETURN
                           ;
                           ; ERRORS
                           ;
000402                     101$:   EMSG$R  M2                      ; ILLEGAL CALL DATA VALUE

000410                     STRHEX: SAVRG   <R3>                    ; SAVE R3
000412  016701  000000G            MOV     .PSTCN,R1               ; GET NUMBER OF CHARACTERS
000416  022701  000040             CMP     #DSCMX,R1               ; TOO MANY CHARACTERS?
000422  103436                     BLO     40$                     ; BR IF YES
000424  110112                     MOVB    R1,(R2)                 ; STORE HEX DIGIT COUNT
000426  016702  000000G            MOV     .PSTPT,R2               ; GET ADDRESS OF SPECIFIED STRING
000432  012746  000001             MOV     #1,-(SP)                ; INIT FLAG (0-LOW BITS,1-HIGH BITS)
000436  112203             10$:    MOVB    (R2)+,R3                ; GET CHARACTER
000440  122703  000106             CMPB    #'F,R3                  ; IS IT A LEGAL HEX VALUE?
000444  103425                     BLO     40$                     ; BR IF NO
000446  122703  000101             CMPB    #'A,R3                  ; IS IT A NUMBER?
000452  101003                     BHI     15$                     ; BR IF YES
000454  162703  000067             SUB     #67,R3                  ; CONVERT ASCII HEX ALPHA TO BINARY
000460  000402                     BR      17$                     ; CONTINUE
000462  162703  000060     15$:    SUB     #60,R3                  ; CONVERT ASCII NUMBER TO BINARY
000466  005716             17$:    TST     (SP)                    ; STORE IN LOW BITS?
000470  001003                     BNE     20$                     ; BR IF NO
000472  150320                     BISB    R3,(R0)+                ; ELSE STORE IN LOW FOUR BITS OF BYTE
000474  005216                     INC     (SP)                    ; SET FLAG TO STORE IN HIGH BITS
000476  000406                     BR      30$                     ; CONTINUE
000500                     20$:    ASL$    4,R3                    ; MOVE TO HIGH FOUR BITS
000510  110310                     MOVB    R3,(R0)                 ; STORE HEX VALUE
000512  005316                     DEC     (SP)                    ; SET FLAG TO STORE IN LOW BITS
000514  005301             30$:    DEC     R1                      ; MORE TO GET?
000516  003347                     BGT     10$                     ; BR IF YES
000520  005226             40$:    INC     (SP)+                   ; CLEAN UP STACK
000522                             RESRG   <R3>                    ; RESTORE R3
000524                             RETURN

```

```
                                        .SBTTL  Parse table for DSN$DF

000054                                  ISTAT$  DSNST,DSNKW

000054                                  STATE$  DSNDF
000054                                    TRAN$   !SKIPSP               ; Skip leading spaces
000054                                  STATE$
000054                                    TRAN$   %DSN$DF%,,,1,SYNERR   ; Flag DSN$DF recognised

000054                                  STATE$
000054                                    TRAN$   !SKIPSP               ; Skip leading spaces
000054                                  STATE$
000054                                    TRAN$   $RAD50,,DSNNOD        ; Parse node name

000054                                  STATE$
000054                                    TRAN$   !END, $EXIT           ; Check for end
000054                                    TRAN$   !OPSEP                ; Skip operand separator
000054                                  STATE$
000054                                    TRAN$   !STRING,,DSNUSR       ; Parse user

000054                                  STATE$
000054                                    TRAN$   !END, $EXIT           ; Check for end
000054                                    TRAN$   !OPSEP                ; Skip operand separator
000054                                  STATE$
000054                                    TRAN$   !STRING,,DSNPAS       ; Parse password

000054                                  STATE$
000054                                    TRAN$   !END, $EXIT           ; Check for end
000054                                    TRAN$   !OPSEP                ; Skip operand separator
000054                                  STATE$
000054                                    TRAN$   !STRING,,DSNACT       ; Parse account

000054                                  STATE$
000054                                    TRAN$   !END,$EXIT            ; Ensure correct termination
                                ;
                                ; Parser subexpressions
                                ;
000054                                  STATE$  OPSEP                   ; Skip separators between operands
000054                                    TRAN$   !SKIPSP               ; Skip trailing spaces
000054                                  STATE$
000054                                    TRAN$   COMMA                 ; Ensure separating comma
000054                                  STATE$
000054                                    TRAN$   !SKIPSP               ; Skip leading spaces
000054                                  STATE$
000054                                    TRAN$   $LAMDA,$EXIT          ; RETURN

000054                                  STATE$  SKIPSP                  ; Skip leading spaces and tabs
000054                                    TRAN$   $BLANK,SKIPSP
000054                                    TRAN$   $LAMDA,$EXIT          ; RETURN

000054                                  STATE$  END                     ; End-of-line test
000054                                    TRAN$   $EOS,$EXIT
000054                                    TRAN$   COMENT,$EXIT
```

```
VFPDST - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:35
Table of contents
```

```
     5-   56    MACRO DEFINITIONS
     6-   78    LOCAL SYMBOL DEFINITIONS
     7-   92    ERROR MESSAGES                                                   ;[TD001]
     8-  107    LOOK FOR DST$DF MACRO
     9-  139    TPARS STATE TABLES
    10-  186    DST$DF ACTION ROUTINES
    13-  252    DSTEND - END OF DESTINATION DESCRIPTOR BLOCK MACROS
    14-  292    CKDSN0 - Check destination name is unique [new format]           ;[TD001]
    15-  328    CKDSN1 - Check destination name is unique [old format]           ;[TD001]
    16-  361    ALDST0 - Allocate/fill destination descriptor [new format]       ;[TD001]
    17-  422    ALDST1 - Allocate/fill destination descriptor [old format]       ;[TD001]
    18-  491    DSTINS - INSERT DESTINATION DESCRIPTOR BLOCK INTO LIST
    19-  525    FNDLEN - Find total length of variable byte fields               ;[TD001]
    20-  550    BCDLEN - Find total length of variable BCD fields                ;[TD001]
    21-  577    MOVAR - Move variable-length byte fields                         ;[TD001]
    22-  606    MOVBCD - Move variable-length BCD fields                         ;[TD001]
    23-  637    MOVLEN - Move length fields                                      ;[TD001]
```

```
                                         .SBTTL  ALDSTO - Allocate/fill destination descriptor [new format]      ;[TD001]
                                 ;+                                                                              ;[TD001]
                                 ; ***   ALDSTO - Allocate/fill destination descriptor [new format]              ;[TD001]
                                 ;                                                                               ;[TD001]
                                 ;                                                                               ;[TD001]
                                 ; INPUTS:                                                                       ;[TD001]
                                 ;       All parsed fields for destination block                                ;[TD001]
                                 ;                                                                               ;**-10
                                 ; OUTPUTS:
                                 ;       CARRY CLEAR - ALLOCATION SUCCESSFUL
                                 ;       CARRY SET - UNSUCCESSFUL ALLOCATION
                                 ;       All registers preserved                                                 ;[TD001]
                                 ;-                                                                              ;[TD001]
000654                           ALDSTO: CALL    $SAVAL                  ; Save all registers                    ;[TD001]
000660                                   SAVMAP                          ; Save current mapping                  ;[TD001]
000664  012701  000032                   MOV     #D$GLEN,R1              ; Fixed length of block                 ;[TD001]
000670  012700  000000G                  MOV     #DSTVR0,R0              ; Get table of variable byte fields     ;[TD001]
000674                                   CALL    FNDLEN                  ; Add length of variable byte fields    ;[TD001]
000700  012700  000000G                  MOV     #DSTVR1,R0              ; Get table of variable BCD fields      ;[TD001]
000704                                   CALL    BCDLEN                  ; Add length of variable BCD fields     ;[TD001]
000710                                   CALL    $ALPOL                  ; Allocate block from network pool      ;[TD001]
000714  103513                           BCS     100$                    ; Branch if unsuccessful allocation     ;[TD001]
000716  010005                           MOV     R0,R5                   ; Save unmapped address of allocation   ;[TD001]
000720                                   CEACC$  R0                      ; Convert to mapped address             ;[TD001]
000730                                   SAVRG   <R0,R1>                 ; Save address and length of allocation ;[TD001]
000734  012702  000000G                  MOV     #$BFR,R2                ; Get address of destination block      ;[TD001]
000740  105022                   15$:    CLRB    (R2)+                   ; Initialize block                      ;[TD001]
000742  005301                           DEC     R1                                                              ;[TD001]
000744  003375                           BGT     15$                                                             ;[TD001]
000746  116767  000000G 000002G          MOVB    DSTPRI,D$PRI+$BFR       ; Set destination priority              ;[TD001]
000754  116767  000000G 000003G          MOVB    DSTOBJ,D$OBJ+$BFR       ; Set object number                     ;[TD001]
000762  016767  000000G 000016G          MOV     DSASLO,D$SLO+$BFR       ; Set low subaddress                    ;[TD001]
000770  016767  000000G 000020G          MOV     DSASHI,D$SHI+$BFR       ; Set high subaddress                   ;[TD001]
000776  012700  000010G                  MOV     #$BFR+D$NOD,R0          ; Address of node name location         ;[TD001]
001002  012701  000006                   MOV     #MXNOID,R1              ; Max node length                       ;[TD001]
001006                                   CALL    SPFILL                  ; Set node name to blanks               ;[TD001]
001012  116701  000000G                  MOVB    RNNL,R1                 ; Actual node length                    ;[TD001]
001016  012702  000010G                  MOV     #$BFR+D$NOD,R2          ; Target area for node name             ;[TD001]
001022  012700  000000G                  MOV     #RNNODE,R0              ; Source of node name                   ;[TD001]
001026                                   CALL    MOVE                    ; Move node name                        ;[TD001]
001032  012700  000000G                  MOV     #DSTVR0,R0              ; Get table of variable byte fields     ;[TD001]
001036  012701  000032G                  MOV     #$BFR+D$GVBL,R1         ; Target area for variable fields       ;[TD001]
001042                                   CALL    MOVAR                   ; Move byte fields                      ;[TD001]
001046  012700  000000G                  MOV     #DSTVR1,R0              ; Get table of variable BCD-fields      ;[TD001]
001052                                   CALL    MOVBCD                  ; Move BCD fields                       ;[TD001]
001056  012700  000000G                  MOV     #DSTVR0,R0              ; Get table of variable byte fields     ;[TD001]
001062  012701  000022G                  MOV     #$BFR+D$DSL,R1          ; Target area for lengths of fields     ;[TD001]
001066                                   CALL    MOVLEN                  ; Set various field lengths             ;[TD001]
001072  116767  000000G 000031G          MOVB    DSARCT,$BFR+D$DTEL      ; Move number of DTE digits             ;[TD001]
001100  116703  000000G                  MOVB    DSCMCT,R3               ; Get number of digits in call mask     ;[TD001]
001104  006203                           ASR     R3                      ; Convert to number of bytes            ;[TD001]
001106  005503                           ADC     R3                                                              ;[TD001]
001110  110367  000030G                  MOVB    R3,$BFR+D$DATL          ; Move number of bytes to block         ;[TD001]
001114                                   RESRG   <R1,R0>                 ; Restore address and length of block   ;[TD001]
001120                                   PUTAD   R0,R1                   ; Write block back to image             ;[TD001]
001134  010501                           MOV     R5,R1                   ; Get unmapped address of block         ;[TD001]
001136                                   CALL    DSTINS                  ; Insert block into list                ;[TD001]
```

```
VFPDST     CREATED BY  MACRO  ON 29-JUN-85 AT 02:36     PAGE 2
SYMBOL CROSS REFERENCE                                  CREF   04.00
SYMBOL  VALUE              REFERENCES

D$VBL      000032          17-475
FMT10    = ****** GX         7-95      7-96      7-97      7-98      7-99     7-100
FMT8     = ****** GX        7-101
FM.10    = 000000           #7-95     #7-96     #7-97     #7-98     #7-99    #7-100
FM.8     = 000000          #7-101
FNDLEN     001626 R        16-378   #19-541
F.RSIZ   = ****** GX      *14-309   *14-314   *15-344   *15-349   *16-415   *17-484   *18-510   *18-516
H$DST      000012          13-263    18-506    18-509
H$D29      000014          13-265    18-509
MAXID    = 000020           #6-88    14-314
MOVAR      001662 R        16-403   #21-594
MOVBCD     001704 R        16-405    17-477   #22-623
MOVE     = ****** GX       16-400    17-459    17-474
MOVLEN     001732 R        16-408   #23-654
MXNOID   = 000006           #6-89    16-395
N$$VCT   = ******          14-313    15-348    16-384    17-443    18-515
PF.XDF   = ****** GX       13-266
RNAL     = ****** GX      *10-200
RNNL     = ****** GX      *10-197    16-397
RNNODE   = ****** GX       16-399
RNPL     = ****** GX      *10-199
RNUL     = ****** GX      *10-198
RTSPC    = ****** GX         7-95      7-96      7-97      7-98      7-99     7-100     7-101
SPACE    = 000040           #6-84
SPFILL   = ****** GX       16-396    17-455    17-470
STORID   = ****** GX       10-193    12-244
SYNERR   = ****** GX       *8-127    *8-130
SYSFDB   = ****** GX      *14-309   *14-314   *15-344   *15-349   *16-415   *17-484   *18-510   *18-516
S$$BAS   = ******            7-95      7-95      7-96      7-96      7-97      7-97      7-98      7-98      7-99
                             7-99     7-100     7-100     7-101     7-101

$ALPHA   = 000022          #9-143
$ALPOL   = ****** GX       16-381    17-440
$ANY     = 000020          #9-143
$BFR     = ****** GX       14-310    14-315    14-318    15-345    15-350    16-386   *16-390   *16-391   *16-392
                          *16-393    16-394    16-398    16-402    16-407   *16-409   *16-413    17-445   *17-449
                          *17-450   *17-451   *17-452    17-453    17-457    17-460    17-468    17-472    17-475
                          *17-478   *17-482    18-513    18-517
$BLANK   = 000006          #9-143
$CAT5B   = ****** GX       17-464
$CEACC   = ****** GX       14-313    15-348    16-384    17-443    18-515
$DBG$    = ******          13-257
$DIGIT   = 000024          #9-143
$DNUMB   = 000014          #9-143
$EOS     = 000012          #9-143
$ERRM7     000204 R        #7-100    13-288
$ERRM9     000236 R        #7-101    13-289
$ERRN4     000000 R         #7-95    10-210
$ERRN5     000032 R         #7-96    11-234
$ERRN6     000062 R         #7-97    11-235
$ERRN7     000110 R         #7-98    12-250
$ERR1T   = ****** GX        8-137
$ERR6M     000132 R         #7-99    13-290
```

```
                                         .SBTTL  DTE$DF ACTION ROUTINES
                                 ;
                                 ; START OF DTE ADDRESS
                                 ;
000052  005067  000000G          DTEAST: CLR     DTEFLG               ; INITIALISE FLAGS (ON/OFF BITS)
000056  012700  000017                   MOV     #DTEALN,R0           ; GET LENGTH OF DTE ADDRESS
000062  012701  000000'                  MOV     #DTEADD,R1           ; GET ADDRESS OF BUFFER
000066  105021                   10$:    CLRB    (R1)+                ; INITIALIZE BUFFER
000070  005300                           DEC     R0                   ; MORE TO INITIALIZE?
000072  003375                           BGT     10$                  ; BR IF YES
000074  105067  000000G                  CLRB    COUNT                ; INITIALIZE DTE DIGIT COUNT
000100  012767  000000' 000000G          MOV     #DTEADD,NEXT         ; START STORING AT BEGINNING OF BUFFER
000106                                   CALL    STRNXT               ; STORE CHARACTER
000112                                   RETURN

                                 ;
                                 ; PROCESS DTE ADDRESS DIGIT
                                 ;
000114  026727  000000G 000016'  DTEDIG: CMP     NEXT,#DTEND          ; IS ADDRESS TOO MANY CHARACTERS?
000122  101003                           BHI     101$                 ; BR IF YES
000124                                   CALL    STRNXT               ; STORE NEXT DIGIT
000130                                   RETURN
                                 ;
                                 ; ERRORS
                                 ;
000132                           101$:   EMSG$R  03                   ; ILLEGAL DTE ADDRESS

                                 ;
                                 ; END OF DTE ADDRESS
                                 ;
000140  004567  000000G          DTEAEN: JSR     R5,$SAVRG            ; SAVE R3-R5
000144  012700  000010                   MOV     #<DTEALN+1>/2,R0     ; GET LENGTH OF PACKED ADDRESS
000150  012704  000000'                  MOV     #DTEADD,R4           ; GET ADDRESS OF DTE ADDRESS BUFFER
000154  012705  000017'                  MOV     #DTEPCK,R5           ; GET ADDRESS OF PACKED ADDRESS BUFFER
000160                                   CALL    PCKBCD               ; PACK ADDRESS IN BCD FORMAT

000164  012700  000027'                  MOV     #DTNTNM,R0           ; Clear DTE name to spaces              ;[TD001]
000170  012701  000006                   MOV     #DTNMMX,R1                                                   ;[TD001]
000174                                   CALL    SPFILL                                                       ;[TD001]
000200  032767  000000G 000000G          BIT     #PF.XDF,$PFLAG       ; Check for extended data format        ;[TD001]
000206  001022                           BNE     20$                  ; If extended, cannot default DTE name  ;[TD001]
                                                                                                              ;[TD001]
000210  016700  000000G                  MOV     $PSIPT,R0            ; POINT TO HOME BLOCK
000214                                   GETRV   R0,#H$LEN            ; READ IN HOME BLOCK
000232  012700  000024G                  MOV     #$BFR+H$NETW,R0      ; POINT TO DEFAULT NET NAME
000236  012701  000027'                  MOV     #DTNTNM,R1           ; POINT TO LOCAL STORAGE OF NAME
000242  012702  000006                   MOV     #DTNMMX,R2           ; GET SIZE OF NAME
000246  112021                   10$:    MOVB    (R0)+,(R1)+          ; SET UP DEFAULT FOR DTE
000250  005302                           DEC     R2                   ; MORE TO SET UP?
000252  003375                           BGT     10$                  ; YES, SET AWAY
                                                                                                              ;[TD001]
000254                           20$:    RETURN                       ; ELSE COMPLETED                        ;[TD001]
```

```
VFPDTE      CREATED BY  MACRO  ON 29-JUN-85 AT 02:37      PAGE 4

SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE               REFERENCES

$ERR07     000120 R          #8-127        15-427
$ERR1T   = ******   GX        9-164
$EXIT    = 000000            #10-170
$FAIL    = 177777            #10-170
$GETAD   = ******   GX        16-457
$GETRV   = ******   GX        12-284       16-449
$GPRM    = ******             10-170
$LAMDA   = 000000            #10-170
$LLCTA   = ******   GX        15-397
$NUMBR   = 000002            #10-170
$PFLAG   = ******   GX        12-280       15-362       15-409
$PSIPT   = ******   GX        12-283       15-417       16-447
$PUTAD   = ******   GX        15-416
$QDTE      000000 RG         #9-148
$RADDR   = ******   GX       *12-284      *15-416      *16-449      *16-457
$RAD50   = 000016            #10-170
$RBLCK   = ******   GX        15-370      *15-421       16-450      *16-467
$RONLY   = ******             10-170       10-170       10-170
$SAVAL   = ******   GX        15-366       16-446       17-492
$SAVRG   = ******   GX        12-271
$SLTMA   = ******   GX        17-504
$SLTNM   = ******   GX        17-503
$STRNG   = 000004            #10-170
$SUBXP   = 000010            #10-170
$$$FLG   = 177777            #10-170
$$$KEY   = 177777            #10-170
.PNUMB   = ******   GX        13-299       13-319
.PNUMH   = ******   GX        13-297       13-317
.PSTCN   = ******   GX        14-331
.PSTPT   = ******   GX        14-334
.TPARS   = ******   GX        9-155
```

```

                                     ;
                                     ; LOW END OF SUBADDRESS RANGE
                                     ;
000352                               PSNSLO: CALL    CHKSUB              ; CHECK FOR VALID SUBADDRESS VALUE
000356  103410                               BCS     101$                ; BR IF INVALID VALUE
000360  010167  000002'                      MOV     R1,PSNLO            ; SAVE SUBADDRESS VALUE
000364  010167  000004'                      MOV     R1,PSNHI            ; DEFAULT IF HIGH OMITTED
000370  052767  000002  000000'              BIS     #HF$DLM,PSNFLG      ; INDICATE SUBADDRESS WAS SPECIFIED
000376                                       RETURN
                                     ;
                                     ; ERRORS
                                     ;
000400                               101$:   EMSG$R  D1                  ; INVALID LOW SUBADDRESS VALUE

                                     ;
                                     ; CHECK FOR VALID SUBADDRESS VALUE
                                     ;
000406  005767  000000G              CHKSUB: TST     .PNUMH              ; LEGAL VALUE
000412  001010                               BNE     10$                 ; BR IF ILLEGAL VALUE
000414  016701  000000G                      MOV     .PNUMB,R1           ; GET SUBADDRESS VALUE
000420  002405                               BLT     10$                 ; BR IF ILLEGAL VALUE
000422  020127  000000G                      CMP     R1,#SUBAMX          ; LEGAL VALUE?
000426  101002                               BHI     10$                 ; BR IF NO
000430  000241                               CLC                         ; INDICATE LEGAL VALUE
000432  000401                               BR      20$                 ; AND EXIT
000434  000261                       10$:    SEC                         ; INDICATE ILLEGAL VALUE
000436                               20$:    RETURN
```

```
                                        .SBTTL  MACRO DEFINITIONS

                                ;
                                ; LIBRARY MACROS
                                ;
                                        .MCALL  EMSG$R,SAVRG,RESRG,PVCDF$,XCBDF$,GETAD,PUTAD,CEACC$
                                        .MCALL  PHBDF$,NTLER$,STATE$,TRAN$,ISTAT$,GETRV
                                        .MCALL  VNPDBG                                                          ;[TD001]
                                ;
                                ; LIBRARY SYMBOLS
                                ;
      000000                            PHBDF$                  ; PSI HOME BLOCK OFFSETS
      000000                            PVCDF$                  ; PVC NAME BLOCK OFFSETS
      000000                            XCBDF$                  ; X25 CIRCUIT CONTROL BLOCKS
                                ;
                                ; LOCAL MACRO DEFINITIONS
                                ;
                                        .MACRO  SAVMAP
                                        MOV     $RBLCK,-(SP)
                                        .ENDM

                                        .MACRO  RESMAP
                                        MOV     (SP)+,$RBLCK
                                        .ENDM
                                ;
                                ; SET UP APR5 MAPPING
                                ;
                                        .MACRO  MAP     BIAS
                                        MOV     BIAS,$RBLCK
                                        .ENDM
                                ;
                                ; SET UP APR5 BIAS
                                ;
                                        .MACRO  BIAS    REG
                                        BIC     #160000,REG     ; CLEAR CURRENT MAPPING BITS
                                        BIS     #120000,REG     ; SET UP APR5 MAPPING
                                        .ENDM
```

```
                                                                  I  1
VFGSVC     CREATED BY  MACRO  ON 29-JUN-85 AT 02:31      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE              REFERENCES

LF.UNL   = 020000           #5-65
LF.X2P   = 000000           #5-65
LN.CLO   = 000000           #5-65
LN.DUM   = 000005           #5-65
LN.LOA   = 000004           #5-65
LN.LOO   = 000003           #5-65
LN.OAU   = 000003           #5-65
LN.OFF   = 000001           #5-65
LN.ON    = 000000           #5-65
LN.OOP   = 000004           #5-65
LN.OPE   = 000001           #5-65
LN.REF   = 000002           #5-65
LN.SER   = 000002           #5-65
LN.STA   = 000017           #5-65
LN.SUB   = 000360           #5-65
LN.TRI   = 000006           #5-65
LTIMR      000004 R         #6-119      *13-433      15-591
L.COST     000015           #5-65
L.CTL      000012           #5-65        10-283
L.CVA      177776           #5-65
L.DDM      000002           #5-65
L.DDS      000004           #5-65
L.DLC      000003           #5-65
L.DLM      000006           #5-65
L.DLS      000010           #5-65
L.FLG      000000           #5-65
L.XRBA     000016           #5-65
L.[illegible]EN   = 000022           #5-65
L.M[illegible]F     000022           #5-65
L.NMST     000020           #5-65
L.NSTA     000014           #5-65
L.OWNR     000021           #5-65
L.UNT      000013           #5-65
MAXWND   = 000007           #6-110       11-324
NEXT       000000 R         #6-117      *12-390      12-397       17-651      *17-652
NODTE      000420 R         #12-374
N$TLC      000100             15-569     15-570       15-571
N$XLEN     000124             15-568
N$$VCT   = ******             13-445     13-474
OWNSVC     000020 R         #6-125       13-467      *14-524       15-561
PCKBCD     001566 R           12-413    #16-612
PKTSVC     000006 R         #6-120      *10-270      *11-312       13-454
PX$DLM   = 000200             15-582
PX$SVC   = 000100             15-582
RCLSVC     000023 R         #6-128      *11-352       13-456
RTMSVC     000012 R         #6-122      *12-364       13-457       13-458
RTSPC    = ******  GX         7-140       7-141        7-142        7-143        7-144        7-145        7-146        7-147        7-148
                              7-149
R$$EIS   = ******             16-623
R$$11D   = ******             16-623
SETTLC     001272 R           13-482    #15-545
SF$INC   = 000002             12-374
```

```
                                        .SBTTL  X2P$DF STATE TABLE
                               ;
                               ;  TPARS STATE TABLES
                               ;
        000216                          ISTAT$  X2PST,X2PKW
                               ;
                               ;  X2P$DF
                               ;
        000216                          STATE$  X2PDF
        000216                           TRAN$   %X2P$DF%,,,1,SYNERR

        000216                          STATE$
        000216                           TRAN$   $RAD50,,X2.DEV          ; DEVICE NAME

        000216                          STATE$
        000216                           TRAN$   <'->

        000216                          STATE$
        000216                           TRAN$   $NUMBR,,X2.CTL          ; CONTROLLER NUMBER

        000216                          STATE$
        000216                           TRAN$   <',>,X2PRM,CHKDEV
        000216                           TRAN$   <'->

        000216                          STATE$
        000216                           TRAN$   $NUMBR,,X2.UNT ; UNIT NUMBER

        000216                          STATE$
        000216                           TRAN$   <',>

        000216                          STATE$   X2PRM
        000216                           TRAN$   $NUMBR,,X2.MWS          ; MAXIMUM WINDOW SIZE

        000216                          STATE$
        000216                           TRAN$   <',>

        000216                          STATE$
        000216                           TRAN$   $NUMBR,,X2.MBS          ; MAXIMUM BLOCK SIZE

        000216                          STATE$
        000216                           TRAN$   <',>

        000216                          STATE$
        000216                           TRAN$   $NUMBR,,X2.MRT          ; MAXIMUM RETRANSMIT COUNT

        000216                          STATE$
        000216                           TRAN$   <',>

        000216                          STATE$
        000216                           TRAN$   $NUMBR,,X2.RTR          ; RETRANSMIT TIMER

        000216                          STATE$
        000216                           TRAN$   <',>

        000216                          STATE$
        000216                           TRAN$   $NUMBR,,X2.HLD          ; HOLDBACK TIMER

        .TITLE  VFGX3P - CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  04-MAR-82
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;



```

```
**FILE**ID**VFIND

VV      VV   FFFFFFFFFF     IIIIII      NN      NN   DDDDDDDD
VV      VV   FFFFFFFFFF     IIIIII      NN      NN   DDDDDDDD
VV      VV   FF               II        NN      NN   DD      DD
VV      VV   FF               II        NN      NN   DD      DD
VV      VV   FF               II        NNNN    NN   DD      DD
VV      VV   FF               II        NNNN    NN   DD      DD
VV      VV   FFFFFFFF         II        NN  NN  NN   DD      DD
VV      VV   FFFFFFFF         II        NN  NN  NN   DD      DD
VV      VV   FF               II        NN    NNNN   DD      DD
VV      VV   FF               II        NN    NNNN   DD      DD
  VV  VV     FF               II        NN      NN   DD      DD       ....
  VV  VV     FF               II        NN      NN   DD      DD       ....
    VV       FF             IIIIII      NN      NN   DDDDDDDD         ....
    VV       FF             IIIIII      NN      NN   DDDDDDDD         ....



LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
VFIND      CREATED BY  MACRO  ON 29-JUN-85 AT 02:33     PAGE 3

SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE           REFERENCES

LS.TST  = 000010        #5-66
LS.UNF  = 020000        #5-66
LS.XIT  = 000011        #5-66
LX.ALI  = 000017        #5-66
LX.CEX  = 000004        #5-66
LX.CIR  = 000034        #5-66
LX.LIN  = 000006        #5-66
LX.LOG  = 000021        #5-66
LX.NOD  = 000015        #5-66
LX.OBJ  = 000023        #5-66
LX.PRO  = 000005        #5-66
N$$VCT  = ******         7-98      8-140     9-188     10-226
OF$INS  = 000001        #5-66
OF$LOG  = 000100        #5-66
OF$OFF  = 000002        #5-66
OF$ON   = 000000        #5-66
OF$SMC  = 000200        #5-66
OP$ADD  = 000200        #5-66
OP$ALL  = 000200        #5-66
OP$CCS  = 000004        #5-66
OP$CIE  = 004000        #5-66
OP$CON  = 000400        #5-66
OP$COP  = 000001        #5-66
OP$COS  = 000100        #5-66
OP$DST  = 000200        #5-66
OP$DTE  = 000040        #5-66
OP$DUP  = 000002        #5-66
OP$EST  = 000100        #5-66
OP$EVE  = 010000        #5-66
OP$FIL  = 000200        #5-66
OP$GRP  = 000100        #5-66
OP$HOS  = 000001        #5-66
OP$HTM  = 000002        #5-66
OP$INA  = 004000        #5-66
OP$INC  = 001000        #5-66
OP$KDS  = 002000        #5-66
OP$KDT  = 000400        #5-66
OP$KEV  = 001000        #5-66
OP$KGR  = 001000        #5-66
OP$KNT  = 010000        #5-66
OP$KSK  = 002000        #5-66
OP$LCT  = 000200        #5-66
OP$LNE  = 000040        #5-66
OP$LOC  = 000001        #5-66     #5-66
OP$LST  = 000002        #5-66
OP$LTM  = 000001        #5-66
OP$LVL  = 000010        #5-66
OP$MAC  = 000010        #5-66
OP$MCO  = 000002        #5-66
OP$MDE  = 000400        #5-66
OP$MLI  = 000004        #5-66
OP$MON  = 000100        #5-66
```

```
                                          .SBTTL  CUG$DF ACTION ROUTINES

                                 ;
                                 ; CUG NAME
                                 ;
000052                           CUNAM:                                                                  ;[TD001]
000052  012700  000000'                  MOV     #CUGNML,R0          ; Get address of CUG buffer           ;[TD001]
000056  012701  000006                   MOV     #CUNMMX,R1          ; Get maximum length for old systems  ;[TD001]
000062                                   CALL    STORID              ; Check CUG name and store in buffer  ;[TD001]
000066  103403                           BCS     101$                ; Carry set if name illegal           ;[TD001]
000070  105067  000032'                  CLRB    CUGFLG              ; Set up default for flags byte       ;[TD001]
000074                                   RETURN                                                            ;**-14
                                 ;
                                 ; ERRORS
                                 ;
000076                           101$:   EMSG$R  N2                  ; ILLLEGAL CUG NAME

                                 ;
                                 ; START OF CUG NUMBER
                                 ;
000104  012700  000004           CUNBST: MOV     #CUGNLN,R0          ; GET LENGTH OF DTE ADDRESS
000110  012701  000024'                  MOV     #CUGNUM,R1          ; POINT TO START OF BUFFER
000114  105021                   10$:    CLRB    (R1)+               ; INITIALIZE BUFFER
000116  005300                           DEC     R0                  ; MORE TO INITIALIZE?
000120  003375                           BGT     10$                 ; BR IF YES
000122  105067  000000G                  CLRB    COUNT               ; INITIALIZE COUNT FIELD
000126  012767  000024' 000000G          MOV     #CUGNUM,NEXT        ; START STORING AT BEGINNING OF BUFFER
000134                                   CALL    STRNXT              ; STORE NEXT CHARACTER
000140                                   RETURN

                                 ;
                                 ; PROCESS CUG NUMBER DIGIT
                                 ;
000142  026727  000000G 000027'  CUGDIG: CMP     NEXT,#CUGEND        ; IS ADDRESS TOO MANY CHARACTERS?
000150  101003                           BHI     101$                ; BR IF YES
000152                                   CALL    STRNXT              ; STORE NEXT CHARACTER
000156                                   RETURN
                                 ;
                                 ; ERRORS
                                 ;
000160                           101$:   EMSG$R  N3                  ; ILLEGAL CUG ADDRESS
```

```
**FILE**ID**VFPDSA








VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD       SSSSSSSS     AAAAAA
VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD       SSSSSSSS     AAAAAA
VV      VV   FF           PP      PP   DD      DD   SS           AA      AA
VV      VV   FF           PP      PP   DD      DD   SS           AA      AA
VV      VV   FF           PP      PP   DD      DD   SS           AA      AA
VV      VV   FF           PP      PP   DD      DD   SS           AA      AA
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD     SSSSSS     AA      AA
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD     SSSSSS     AA      AA
VV      VV   FF           PP           DD      DD           SS   AAAAAAAAAA
VV      VV   FF           PP           DD      DD           SS   AAAAAAAAAA
  VV  VV     FF           PP           DD      DD           SS   AA      AA       ....
  VV  VV     FF           PP           DD      DD           SS   AA      AA       ....
    VV       FF           PP           DDDDDDDD     SSSSSSSS     AA      AA       ....
    VV       FF           PP           DDDDDDDD     SSSSSSSS     AA      AA       ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
    343         000001                          .END
```

```
                                        ;
                                        ; Parse a string
                                        ;
                                        ; String can be sequence of alphanumeric characters terminated by non alphan.
                                        ;             or sequence of characters in quotes ("...")
                                        ;             or sequence of characters in angle brackets (<...>)
                                        ;             or sequence of characters in quotes and angle brackets (<"...">)
                                        ;
000054                                          STATE$  STRING                  ; Parse quoted/unquoted string
000054                                            TRAN$   BRA,BRAST,CLRSTR      ; Check for bracketed string
000054                                            TRAN$   DQUOTE,QST,CLRSTR     ; Check for quoted string
000054                                            TRAN$   $LAMDA,NQST,CLRSTR    ; Must be unbracketed, unquoted

000054                                          STATE$  BRAST                   ; String in brackets
000054                                            TRAN$   DQUOTE,QBRAST         ; Check for quoted, bracketed
000054                                            TRAN$   $LAMDA
000054                                          STATE$  BNQST                   ; Bracketed unquoted string
000054                                            TRAN$   KET,$EXIT             ; Exit on end bracket
000054                                            TRAN$   $EOS,$EXIT            ; Exit at end-of-string
000054                                            TRAN$   $ANY,BNQST,STORCH     ; Accept any other character
000054                                          STATE$  QBRAST                  ; Quoted, bracketed string
000054                                            TRAN$   !QST                  ; Get quoted string
000054                                          STATE$
000054                                            TRAN$   KET,$EXIT             ; Ensure closing bracket

000054                                          STATE$  QST                     ; Quoted string
000054                                            TRAN$   DQUOTE,DBLQT          ; If quote, check for double quote
000054                                            TRAN$   $ANY,QST,STORCH       ; Accept any other character
000054                                          STATE$  DBLQT
000054                                            TRAN$   DQUOTE,QST,STORCH     ; Recognise two quotes as quote char
000054                                            TRAN$   $LAMDA,$EXIT          ; Exit on single quote

000054                                          STATE$  NQST                    ; Unquoted string
000054                                            TRAN$   $ALPHA,NQST,STORCH    ; Accept alpha-numerics only
000054                                            TRAN$   $DIGIT,NQST,STORCH
000054                                            TRAN$   $LAMDA,$EXIT          ; Exit on anything else

000054                                          STATE$                          ; End of state tables
```

```
        .TITLE  VFPDST - INITIAL CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/
        .ENABL  LC                                                      ;[TD001]
                                                                        ;**-1
;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-82
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
    418 001142  000241              CLC                                 ; Indicate block allocated                  ;[TD001]
    419 001144                100$: RESMAP                              ; Restore previous mapping                  ;[TD001]
    420 001150                      RETURN                                                                          ;[TD001]
```

```
VFPDST     CREATED BY  MACRO  ON 29-JUN-85 AT 02:36      PAGE 3
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE             REFERENCES

$EXIT    = 000000          #9-143
$FAIL    = 177777          #9-143
$GETAD   = ******  GX       14-314      15-349      18-516
$GETRV   = ******  GX       14-309      15-344      18-510
$GPRM    = ******           9-143
$LAMDA   = 000000          #9-143
$NUMBR   = 000002          #9-143
$PFLAG   = ******  GX       13-266
$PSIPT   = ******  GX       13-260      18-505
$PUTAD   = ******  GX       16-415      17-484
$QDST      000000 RG       #8-120
$RADDR   = ******  GX      *14-309     *14-314     *15-344     *15-349     *16-415     *17-484     *18-510     *18-516
$RAD50   = 000016          #9-143
$RBLCK   = ******  GX       14-308     *14-325      15-343     *15-358      16-375     *16-419      17-436     *17-488
$RONLY   = ******           9-143       9-143       9-143
$SAVAL   = ******  GX       15-342      16-374      17-435
$STRNG   = 000004          #9-143
$SUBXP   = 000010          #9-143
$XLINK   = ******  GX       18-522
$$$FLG   = 177777          #9-143
$$$KEY   = 177777          #9-143
.PNUMB   = ******  GX       11-225      11-227
.PNUMH   = ******  GX       11-223
.TPARS   = ******  GX       8-128
```

```
                                         ;
                                         ; HASH TABLE SIZE
                                         ;
 000256  005767  000000G                 HSHTBL: TST     .PNUMH                  ; LEGAL HASH TABLE SIZE VALUE?
 000262  001016                                  BNE     101$                    ; BR IF NO
 000264  016700  000000G                         MOV     .PNUMB,R0               ; GET HASH TABLE SIZE
 000270  012701  000040                          MOV     #HSHMN,R1               ; GET MINIMUM SIZE OF HASH TABLE
 000274  020001                          10$:    CMP     R0,R1                   ; IS THIS A LEGAL SIZE?
 000276  001405                                  BEQ     20$                     ; BR IF YES
 000300  006301                                  ASL     R1                      ; HASH TABLE SIZE MUST BE POWER OF TWO
 000302  020127  001000                          CMP     R1,#HSHMX               ; IS THIS A LEGAL SIZE?
 000306  101004                                  BHI     101$                    ; BR IF NO
 000310  000771                                  BR      10$                     ; CHECK FOR LEGAL VALUE
 000312  010067  000000G                 20$:    MOV     R0,HSHSZ                ; SAVE HASH TABLE SIZE
 000316                                          RETURN
                                         ;
                                         ; ERRORS
                                         ;
 000320                                  101$:   EMSG$R  04                      ; ILLEGAL HASH TABLE SIZE

                                         ;
                                         ; COUNTER TIMER VALUE
                                         ;
 000326  005767  000000G                 CNTTIM: TST     .PNUMH                  ; LEGAL COUNTER TIMER VALUE?
 000332  001004                                  BNE     101$                    ; BR IF NO
 000334  016767  000000G 000000G                 MOV     .PNUMB,CTIM             ; SAVE COUNTER TIMER VALUE
 000342                                          RETURN
                                         ;
                                         ; ERRORS
                                         ;
 000344                                  101$:   EMSG$R  05                      ; ILLEGAL COUNTER TIMER VALUE

```

```
VFPDTE    CREATED BY  MACRO  ON 29-JUN-85 AT 02:37     PAGE 5

MACRO CROSS REFERENCE                                  CREF   04.00

MACRO NAME      REFERENCES

CALL               9-155    12-253    12-261    12-275    12-279    12-284    15-350    15-355    15-366    15-367
                  15-372    15-413    15-416    15-420    16-446    16-449    16-456    16-457    17-492    17-497
CEACC$             #5-61    15-372    16-456
DBGTP$           #10-170   #10-182   #10-201   #10-205   #10-209   #10-216   #11-234
DTEDF$             #5-61      5-67
EMSG$R             #5-61     9-164    12-266    13-312    13-324    14-344    15-426    15-427    15-428    15-429
GETAD              #5-61    16-457
GETRV              #5-62    12-284    16-449
ISTAT$             #5-62    10-170
MTRAN$           #10-170
NTLER$             #5-62     8-125     8-126     8-127     8-128     8-129     8-130
PHBDF$             #5-62      5-68
PUTAD              #5-61    15-416
RAND             #12-284    12-284   #15-416    15-416   #16-449    16-449   #16-457    16-457
RESMAP             #5-77    15-421    16-467
RESRG              #5-61
RETURN             9-159    12-254    12-262    12-292    13-308    13-320    15-422    16-469    17-521    18-542
SAVMAP             #5-73    15-370    16-450
SAVRG              #5-61
SLTDF$             #5-61      5-69
STATE$             #5-62    10-174   #10-177   #10-180   #10-184   #10-187   #10-190   #10-193   #10-196   #10-199
                 #10-203   #10-207   #10-211   #10-214    11-223   #11-226   #11-229   #11-232   #11-236   #11-239
TRAN$              #5-62   #10-175   #10-178   #10-181   #10-182   #10-185   #10-188   #10-191   #10-194   #10-197
                 #10-200   #10-201   #10-204   #10-205   #10-208   #10-209   #10-212   #10-215   #10-216   #11-224
                 #11-227   #11-230   #11-233   #11-234   #11-237
VNPDBG             #5-63     9-148
```

```
                                        ;
                                        ; HIGH END OF SUBADDRESS RANGE
                                        ;
                                                .ENABL  LSB
000440                                  PSNSHI: CALL    CHKSUB                  ; CHECK FOR VALID SUBADDRESS VALUE
000444  103530                                  BCS     101$                    ; BR IF INVALID VALUE
000446  020167  000002'                         CMP     R1,PSNLO                ; HIGH MUST NOT BE LESS THAN LOW
000452  002525                                  BLT     101$                    ; BR IF ERROR
000454  010167  000004'                         MOV     R1,PSNHI                ; SAVE SUBADDRESS VALUE
000460  052767  000002  000000'                 BIS     #HF$DLM,PSNFLG          ; INDICATE SUBADDRESS WAS SPECIFIED
                                        ;
                                        ; End of PSN$DF - allocate homeblock                                          ;[TD001]
                                        ;                                                                             ;[TD001]
000466                                  PSNEND: CALL    $SAVAL                  ; Save registers                      ;[TD001]
000472  012701  000042                          MOV     #H$LEN,R1               ; Get length of V2.0 PSI home block   ;[TD001]
000476  032767  000000G 000000G                 BIT     #PF.XDF,$PFLAG          ; Check for extended data             ;[TD001]
000504  001007                                  BNE     5$                      ; Branch if extended                  ;[TD001]
000506  032767  000004  000000'                 BIT     #HF$HOS,PSNFLG          ; Check for non-extended host         ;[TD001]
000514  001405                                  BEQ     10$                     ; Branch if not host                  ;[TD001]
000516  012701  000044                          MOV     #H$HLEN,R1              ; Length of host PSI home block       ;[TD001]
000522  000402                                  BR      10$                                                           ;[TD001]
000524  012701  000104                  5$:     MOV     #H$GLEN,R1              ; Length of gateway/extended PHB      ;[TD001]
000530                                  10$:    CALL    $ALL16                  ; Allocate home block from DSR        ;[TD001]
000534  103477                                  BCS     102$                    ; Branch if unable to allocate it     ;[TD001]
000536                                          SAVRG   <R0,R1>                 ; Save home block address and length  ;[TD001]
000542  012702  000000G                         MOV     #$BFR,R2                ; Get address of home block           ;[TD001]
000546  105022                          20$:    CLRB    (R2)+                   ; Initialize home block               ;[TD001]
000550  005301                                  DEC     R1                      ; More to initialize?                 ;[TD001]
000552  003375                                  BGT     20$                     ; Branch if yes                       ;[TD001]
000554  012700  000007'                         MOV     #PSNNAM,R0              ; Source of PSN name                  ;[TD001]
000560  012701  000006                          MOV     #PNAMMX,R1              ; Length of old-style PSN name        ;[TD001]
000564  012702  000024G                         MOV     #$BFR+H$NETW,R2         ; Target for old-style PSN name       ;[TD001]
000570  032767  000000G 000000G                 BIT     #PF.XDF,$PFLAG          ; Check for extended format           ;[TD001]
000576  001404                                  BEQ     30$                     ; Use old-style values if not extended ;[TD001]
000600  012701  000020                          MOV     #MAXID,R1               ; Length of extended PSN name         ;[TD001]
000604  012702  000050G                         MOV     #$BFR+H$GNAM,R2         ; Target for extended PSN name        ;[TD001]
000610  112022                          30$:    MOVB    (R0)+,(R2)+             ; Copy PSN name                       ;[TD001]
000612  005301                                  DEC     R1                                                            ;[TD001]
000614  003375                                  BGT     30$                                                           ;[TD001]
000616  016767  000000G 000022G                 MOV     PORTNO,$BFR+H$NPT       ; Copy number of ports                ;[TD001]
000624  016767  000000' 000000G                 MOV     PSNFLG,$BFR+H$FLG       ; Copy homeblock flags                ;[TD001]
000632  032767  000002  000000'                 BIT     #HF$DLM,PSNFLG          ; If DLM, copy subaddress range       ;[TD001]
000640  016767  000002' 000032G                 MOV     PSNLO,$BFR+H$LOTS       ; Low end of range                    ;[TD001]
000646  016767  000004' 000034G                 MOV     PSNHI,$BFR+H$HITS       ; High end of range                   ;[TD001]
000654                                  40$:    RESRG   <R1,R0>                 ; Restore allocation length/address   ;[TD001]
000660                                          PUTRC   R0,R1                   ; Write homeblock to system image     ;[TD001]
000674  010067  000000G                         MOV     R0,$PSIPT               ; Set pointer to homeblock            ;[TD001]
                                        ;                                                                             ;[TD001]
                                        ; Allocate Port Table                                                         ;[TD001]
                                        ;                                                                             ;[TD001]
000700                                          CALL    ALLPRT                  ; Attempt to allocate table           ;[TD001]
000704  103416                                  BCS     103$                    ; Branch if allocation error          ;[TD001]
                                        ;                                                                             ;[TD001]
                                        ; Allocate PSV mapping tables                                                 ;[TD001]
                                        ;                                                                             ;[TD001]
000706  032767  000000G 000000G                 BIT     #PF.PSV,$PFLAG          ; Check for PSV module                ;[TD001]
000714  001403                                  BEQ     70$                     ; Branch if not present               ;[TD001]
```

```
    94                                        .SBTTL  LOCAL SYMBOL DEFINITIONS
    95
    96                                  ;
    97                                  ; LOCAL SYMBOL DEFINITIONS
    98                                  ;
    99          000040                        SPACE = 40              ; ASCII SPACE
   100          000006                        PVNAMX = 6              ; Maximum length of circuit id [old format]      ;[TD001]
   101          000020                        MAXID = 16.             ; Max length of net. mgmt. 'id-string'           ;[TD001]
```

```
                                                                      J   1
VFGSVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:31      PAGE 3
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

SF.ACT  = 000200            #5-65
SF.ENA  = 000100            #5-65
SF.LPB  = 000004            #5-65
SF.MFL  = 000040            #5-65
SF.PAC  = 000020            #5-65
SF.REA  = 000010            #5-65
SF.SER  = 000001            #5-65
SF.SVC  = 000002            #5-65
SF.UNL  = 000040            #5-65
SLTA      000016 R          #6-124     *13-429     15-546
STASVC    000014 R          #6-123     *10-295     13-430     13-453     15-553
STRNXT    001646 R           12-391     12-399    #17-648
SVCDF     000000 R           8-167
SVCKW     000000 RG          8-169     #9-187
SVCST     000000 RG         #9-187
SVCTL     000114 R          #10-277
SVDTED    000476 R          #12-397
SVDTEE    000522 R          #12-409
SVDTES    000440 RG         #12-384
SVFLAG    000316 R          #11-336
SVNAM     000052 R          #10-263
SVOWNR    000550 R          #13-419
SVPKT     000236 R          #11-310
SVRCLL    000342 R          #11-348
SVRRTM    000374 R          #12-362
SVSTA     000164 R          #10-291
SVWND     000262 R          #11-322
SYNERR  = ******   GX       *8-172     *8-175     *10-285    *10-298
SYSFDB  = ******   GX       *13-468    *13-470    *13-475    *15-568    *15-573
S$ADSZ    000007            *13-460
S$CTL     000002            *13-452
S$DTE     000020             13-462
S$FLG     000010            *13-455
S$LEN     000032             13-439     13-468
S$OWNR    000030            *13-467
S$PKSZ    000004            *13-454
S$RCL     000013            *13-456
S$STA     000003            *13-453
S$TIM     000014            *13-457
S$TMR     000016            *13-458
S$WND     000006            *13-459
S$$BAS  = ******             7-140      7-140      7-141      7-141      7-142      7-142      7-143      7-143      7-144
                             7-144      7-145      7-145      7-146      7-146      7-147      7-147      7-148      7-148
                             7-149      7-149
S.COST    0000C             #5-65
S.FLG     000000            #5-65
S.LEN     000004            #5-65
S.NMST    000002            #5-65
S.OWNR    000003            #5-65
T$FLAG    000044            #5-62      *15-582
T$LIF     000013            #5-62
T$LIFL    000013            #5-62
```

```
                                                                      K   1
```

```
    189 000216                                      STATE$
    190 000216                                       TRAN$  <';>,$EXIT
    191 000216                                       TRAN$  $EOS,$EXIT
    192
    193 000216                                      STATE$
```

```
     55                                         .SBTTL  MACRO DEFINITIONS
     56
     57                                 ;
     58                                 ; LIBRARY MACROS
     59                                 ;
     60                                         .MCALL  ISTAT$,STATE$,TRAN$,EMSG$R,NTLER$
```

```
VFIND - VNP FIND ENTITY ROUTINE MACRO V05.03b Saturday 29-Jun-85 02:33
Table of contents
```

```
      5-    54      MACRO DEFINTIONS
      6-    70      LOCAL DATA
      7-    76      $FNOBJ - FIND OBJECT
      8-   109      $FNALI - FIND ALIAS
      9-   161      $FNADD - FIND REMOTE BY ADDRESS
     10-   197      $FNNAM - FIND REMOTE BY NAME
```

```
VFIND      CREATED BY  MACRO  ON 29-JUN-85 AT 02:33      PAGE 4
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL    VALUE              REFERENCES

OP$MST    = 000020           #5-66
OP$NAM    = 000004           #5-66        #5-66        #5-66
OP$NET    = 004000           #5-66
OP$NLI    = 000002           #5-66
OP$NOD    = 000020           #5-66
OP$OUT    = 002000           #5-66
OP$OWN    = 000020           #5-66
OP$PAR    = 000010           #5-66
OP$PRI    = 000010           #5-66
OP$RET    = 010000           #5-66
OP$RPA    = 000010           #5-66
OP$SCO    = 000001           #5-66
OP$SEG    = 000400           #5-66
OP$SER    = 000400           #5-66
OP$SIN    = 000010           #5-66
OP$STA    = 000004           #5-66
OP$TPA    = 000020           #5-66
OP$TRI    = 000040           #5-66
OP$UCS    = 000020           #5-66
OP$USE    = 000002           #5-66
OP$VEC    = 000040           #5-66
OP$VER    = 000040           #5-66
OP$XAC    = 000001           #5-66
OP$XPR    = 000002           #5-66
OP$X5S    = 000004           #5-66
OP$X9S    = 000010           #5-66
O.LEN       000012            7-99
O.TYP       000002            7-102
QF$ACT    = 000001           #5-66
QF$ALL    = 000002           #5-66         8-142
QF$KNO    = 000004           #5-66         7-100        8-144
QF$LOO    = 000010           #5-66
RF.LOO    = 100000           #5-64
RQB       = ******    GX      8-153
R.ADD       000010           #5-64         9-190
R.FLAG      000012           #5-64
R.LEN       000014           #5-64         9-189        10-227
R.LNK       000000           #5-64
R.NAM       000002           #5-64         10-229       10-231       10-233
SYSFDB    = ******    GX     *7-99        *8-127       *8-128       *8-129       *8-141       *9-175       *9-176       *9-177       *9-189
                             *10-213      *10-214      *10-215      *10-227

TY$CHA    = 000400           #5-66
TY$EVE    = 001000           #5-66
TY$STA    = 002000           #5-66
TY$SUM    = 000000           #5-66
WC.CNT    = 000400           #5-66
WC.EVE    = 004000           #5-66
WC.TRI    = 002000           #5-66
WC.UNT    = 001000           #5-66
$BFR      = ******    GX     *7-91         7-95         7-102        8-129       *8-133        8-137        8-147        8-149        8-151
                              8-153        9-177       *9-181        9-185        9-190        10-215      *10-219       10-223       10-229
                              10-231       10-233
```

```
                                          ;
                                          ; END OF CUG ADDRESS
                                          ;
 000166  004567  000000G                  CUNBEN: JSR     R5,$SAVRG               ; SAVE R3-R5
 000172  012700  000002                           MOV     #<CUGNLN/2>,R0          ; GET LENGTH OF PACKED CUG ADDRESS
 000176  116704  000000G                          MOVB    COUNT,R4                ; GET COUNT
 000202  062704  000020'                          ADD     #CUGNUM-4,R4            ; POINT TO NUMBER TO CONVERT
 000206  012705  000030'                          MOV     #CUGPCK,R5              ; POINT TO BUFFER TO STORE BCD ADDRESS
 000212  112767  000004  000000G                  MOVB    #4,COUNT                ; ALWAYS CONVERT FOUR DIGITS
 000220                                           CALL    PCKBCD                  ; PACK CUG ADDRESS IN BCD FORMAT
 000224                                           RETURN
```

Table of contents

```
     5-   56    MACRO DEFINITIONS
     6-   64    LOCAL SYMBOL DEFINITIONS
     7-   72    ERROR MESSAGES
     8-   88    LOOK FOR DSA$DF MACRO
     9-  124    TPARS STATE TABLES
    10-  166    DSC$DF STATE TABLES
    11-  191    DSA$DF ACTION ROUTINES
    14-  286    DSC$DF ACTION ROUTINES
```

```
VFPDSA - INITIAL CONFIG FILE SC MACRD V05.03b Saturday 29-Jun-85 02:34  Page 14-2
Symbol table

A$$CHK= 000000        DSADF   000000R   003 G$$WRD= 000000        PCKBCD= ****** GX     $EDS  = 000012
A$$CPS= 000000        DSAKW   000000RG  004 I$$RAR= 000000        P$$P45= 000000        $ERRM0  000106R   002
A$$PRI= 000000        DSARCT= ****** GX     I$$RDN= 000000        P$$WRD= 000000        $ERRM2  000136R   002
A$$TRP= 000000        DSARDP= ****** GX     K$$CNT= 177546        Q$$DPT= 000010        $ERRM3  000166R   002
CALVAL  000152R   003 DSARDT= ****** GX     K$$CSR= 177546        RTSPC = ****** GX     $ERRN2  000000R   002
CERR  = ****** GX     DSASHI= ****** GX     K$$LDC= 000000        R$$DER= 000000        $ERRN8  000022R   002
CFGBF = ****** GX     DSASLO= ****** GX     K$$TPS= 000074        R$$K11= 000001        $ERRN9  000056R   002
CFGSZ = ****** GX     DSAST   000000RG  003 LD$LP = 000000        R$$SND= 000000        $ERR1T= ****** GX
CFLIN = ****** GX     DSCDF   000130R   003 L$$ASG= 000000        R$$11M= 000000        $EXIT = 000000
CHKSB   000254R       DSCMCT= ****** GX     L$$DRV= 000000        SHI     000102R   003 $FAIL = 177777
COUNT = ****** GX     DSCMSK= ****** GX     L$$P11= 000001        SLO     000062R   003 $LAMDA= 000000
CUNMMX= 000006        DSCMX = 000040        L$$11R= 000000        STORID= ****** GX     $NUMBR= 000002
C$$CKP= 000000        DSCVAL= ****** GX     M$$CRB= 000124        STRHEX  000410R       $QDSA   000000RG
C$$ORE= 000400        DSCVCT= ****** GX     M$$CRX= 000000        STRNXT= ****** GX     $QDSC   000006RG
C$$RSH= 177564        D$$BUG= 177514        M$$FCS= 000000        SUBAMX= ****** GX     $RAD50= 000016
DACUG   000122R   003 D$$ISK= 000000        M$$MGE= 000000        SYNERR= ****** GX     $SAVRG= ****** GX
DACUNM  000306R       D$$L11= 000001        M$$NET= 000000        S$$WRG= 000000        $STRNG= 000004
DARDEN= ****** GX     D$$YNC= 000000        M$$OVR= 000000        S$$YSZ= 007600        $SUBXP= 000010
DARDND  000142R       D$$YNM= 000000        NEXT  = ****** GX     T$$KMG= 000000        $$$FLG= 177777
DARDNO  000116R       END     000160R   003 N$$ACC= 000001        T$$MIN= 000000        $$$KEY= 000001
DARDNX  000032R   003 E$$XPR= 000000        N$$BUF= 000001        V$$CTR= 001000        $$$STA= 000000
DARDST  000060R       FMT10 = ****** GX     N$$LDV= 000001        X$$DBT= 000000        $$$TMP= 000007R   005
DARMLN= ****** GX     FM.10 = 000000        N$$MCP= 000001        $ALPHA= 000022        .PNUMB= ****** GX
DASSHI  000224R       F$$LVL= 000001        N$$MLL= 000001        $ANY  = 000020        .PNUMH= ****** GX
DASSLO  000176R       G$$TPP= 000000        N$$MDV= 000010        $BLANK= 000006        .PSTCN= ****** GX
DCMSK   000334R       G$$TSS= 000000        N$$NCT= 000001        $DIGIT= 000024        .PSTPT= ****** GX
DCVAL   000362R       G$$TTK= 000000        N$$PEM= 000001        $DNUMB= 000014        .TPARS= ****** GX
DSACGL= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000526    001   (RW,I,LCL,REL,CON)
DATA    000216    002   (RW,D,LCL,REL,CON)
$STATE  000170    003   (RW,D,LCL,REL,CON)
$KTAB   000004    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12592 Words  ( 50 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:38.14
SY:VFPDSA.V2,[132,134]VFPDSA/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPDSA
```

```
                                        .SBTTL  Syntax action routines

                                ;+
                                ; *** DSNNOD - Check node name
                                ;
                                ; INPUTS
                                ;       .PSTPT - Pointer to parsed string
                                ;       .PSTCN - Number of characters in string
                                ;
                                ; OUTPUTS
                                ;       Node name checked and stored
                                ;       Error condition if invalid node name
                                ;-
000054                          DSNNOD:
000054  016701  000000G                 MOV     .PSTCN,R1
000060  020127  000006                  CMP     R1,#MXNOID              ; Check if node name too long
000064  101011                          BHI     101$                    ; Branch if error
000066  110167  000000G                 MOVB    R1,RNNL                 ; Move node length to data area
000072  016700  000000G                 MOV     .PSTPT,R0               ; Move node name to data area
000076  012702  000000G                 MOV     #RNNODE,R2
000102                                  CALL    MOVE
000106                                  RETURN

000110                          101$:   EMSG$R  MC                      ; Illegal destination access data
```

```
     56                                                  .SBTTL  MACRO DEFINITIONS
     57
     58                                          ;
     59                                          ; LIBRARY MACROS
     60                                          ;
     61                                                  .MCALL  EMSG$R,DSTDF$,SAVRG,RESRG,PHBDF$,NTLER$
     62                                                  .MCALL  ISTAT$,STATE$,TRAN$,CEACC$,GETAD,GETRV,PUTAD
     63                                                  .MCALL  VNPDBG                                                          ;[TD001]
     64
     65 000000                                           DSTDF$                  ; DESTINATION DESCRIPTOR BLOCK OFFSETS
     66 000000                                           PHBDF$                  ; PSI HOME BLOCK OFFSETS
     67                                          ;
     68                                          ; LOCAL MACRO DEFINITIONS
     69                                          ;
     70                                                  .MACRO  SAVMAP
     71                                                  MOV     $RBLCK,-(SP)
     72                                                  .ENDM
     73
     74                                                  .MACRO  RESMAP
     75                                                  MOV     (SP)+,$RBLCK
     76                                                  .ENDM
```

```
                                           .SBTTL  ALDST1 - Allocate/fill destination descriptor [old format]     ;[TD001]
                                  ;+                                                                               ;[TD001]
                                  ; ***    ALDST1 - Allocate/fill destination descriptor [old format]              ;[TD001]
                                  ;                                                                                ;[TD001]
                                  ;                                                                                ;[TD001]
                                  ; INPUTS:                                                                        ;[TD001]
                                  ;        All parsed fields for destination block                                 ;[TD001]
                                  ;                                                                                ;[TD001]
                                  ; OUTPUTS:                                                                       ;[TD001]
                                  ;        CARRY CLEAR - ALLOCATION SUCCESSFUL                                     ;[TD001]
                                  ;        CARRY SET - UNSUCCESSFUL ALLOCATION                                     ;[TD001]
                                  ;        All registers preserved                                                 ;[TD001]
                                  ;-                                                                               ;[TD001]
001152                            ALDST1:  CALL    $SAVAL                  ; Save all registers                    ;[TD001]
001156                                     SAVMAP                          ; Save current mapping                  ;[TD001]
001162  012701   000032                    MOV     #D$FLEN,R1              ; Fixed length of block                 ;[TD001]
001166  012700   000000G                   MOV     #DSTVR1,R0              ; Get table of variable BCD fields      ;[TD001]
001172                                     CALL    BCDLEN                  ; Add length of variable BCD fields     ;[TD001]
001176                                     CALL    $ALPOL                  ; Allocate block from network pool      ;[TD001]
001202  103535                             BCS     100$                    ; Branch if unsuccessful allocation     ;[TD001]
001204  010005                             MOV     R0,R5                   ; Save unmapped address of allocation   ;[TD001]
001206                                     CEACC$  R0                      ; Convert to mapped address             ;[TD001]
001216                                     SAVRG   <R0,R1>                 ; Save address and length of allocation ;[TD001]
001222  012702   000000G                   MOV     #$BFR,R2                ; Get address of destination block      ;[TD001]
001226  105022                    15$:     CLRB    (R2)+                   ; Initialize block                      ;[TD001]
001230  005301                             DEC     R1                                                              ;[TD001]
001232  003375                             BGT     15$                                                             ;[TD001]
001234  116767   000000G 000002G           MOVB    DSTPRI,D$PRI+$BFR       ; Set destination priority              ;[TD001]
001242  116767   000000G 000003G           MOVB    DSTOBJ,D$OBJ+$BFR       ; Set object number                     ;[TD001]
001250  016767   000000G 000016G           MOV     DSASLO,D$SLO+$BFR       ; Set low subaddress                    ;[TD001]
001256  016767   000000G 000020G           MOV     DSASHI,D$SHI+$BFR       ; Set high subaddress                   ;[TD001]
001264  012700   000010G                   MOV     #$BFR+D$DST,R0          ; Address of destination name location  ;[TD001]
001270  012701   000006                    MOV     #DSNAMX,R1              ; Max destination length                ;[TD001]
001274                                     CALL    SPFILL                  ; Set destination name to blanks        ;[TD001]
001300  116701   000000G                   MOVB    DSTNML,R1               ; Actual destination length             ;[TD001]
001304  012702   000010G                   MOV     #$BFR+D$DST,R2          ; Target area for destination name      ;[TD001]
001310  012700   000000G                   MOV     #DSTNAM,R0              ; Source of destination name            ;[TD001]
001314                                     CALL    MOVE                    ; Move destination name                 ;[TD001]
001320  012703   000004G                   MOV     #$BFR+D$NAM,R3          ; Address target for task name          ;[TD001]
001324  116704   000000G                   MOVB    DSTTKL,R4               ; Get length of task name               ;[TD001]
001330  012700   000000G                   MOV     #DSTTSK,R0              ; Address space-filled task name        ;[TD001]
001334  012701   000001           25$:     MOV     #1,R1                   ; Allow periods in task name            ;[TD001]
001340                                     CALL    $CAT5B                  ; Convert name to RAD50                 ;[TD001]
001344  010123                             MOV     R1,(R3)+                ; Store task name                       ;[TD001]
001346  162704   000003                    SUB     #3,R4                   ; Check if any more conversion needed   ;[TD001]
001352  003370                             BGT     25$                                                             ;[TD001]
001354  012700   000022G                   MOV     #$BFR+D$CUGN,R0         ; Space-fill CUG name                   ;[TD001]
001360  012701   000006                    MOV     #CUNMMX,R1                                                      ;[TD001]
001364                                     CALL    SPFILL                                                          ;[TD001]
001370  116701   000000G                   MOVB    DSACGL,R1               ; Actual length of CUG name             ;[TD001]
001374  012702   000022G                   MOV     #$BFR+D$CUGN,R2         ; Target for CUG name                   ;[TD001]
001400  012700   000000G                   MOV     #DSACUG,R0              ; Source of CUG name                    ;[TD001]
001404                                     CALL    MOVE                    ; Move CUG name                         ;[TD001]
001410  012701   000032G                   MOV     #$BFR+D$VBL,R1          ; Target area for variable fields       ;[TD001]
001414  012700   000000G                   MOV     #DSTVR1,R0              ; Get table of variable BCD-fields      ;[TD001]
001420                                     CALL    MOVBCD                  ; Move BCD fields                       ;[TD001]
001424  116767   000000G 000031G           MOVB    DSARCT,$BFR+D$DTEL      ; Move number of DTE digits             ;[TD001]
```

```
VFPDST     CREATED BY  MACRO  ON 29-JUN-85 AT 02:36     PAGE 4
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

CALL             8-128     10-193    12-244    13-271    13-273    13-279    13-281    14-309    14-313    14-314
                15-342     15-344    15-348    15-349    16-374    16-378    16-380    16-381    16-384    16-396
                16-400     16-403    16-405    16-408    16-415    16-417    17-435    17-439    17-440    17-443
                17-455     17-459    17-464    17-470    17-474    17-477    17-484    17-486    18-510    18-515
                18-516     18-522
CEACC$          #5-62      14-313    15-348    16-384    17-443    18-515
DBGTP$          #9-143    #9-153    #9-175    #9-182
DSTDF$          #5-61       5-65
EMSG$R          #5-61       8-137    10-210    11-234    11-235    12-250    13-288    13-289    13-290
GETAD           #5-62      14-314    15-349    18-516
GETRV           #5-62      14-309    15-344    18-510
ISTAT$          #5-62       9-143
MTRAN$          #9-143
NTLER$          #5-61       7-95      7-96      7-97      7-98      7-99      7-100     7-101
PHBDF$          #5-61       5-66
PUTAD           #5-62      16-415    17-484
RAND            #14-309    14-309   #14-314    14-314   #15-344    15-344   #15-349    15-349   #16-415    16-415
                #17-484    17-484   #18-510    18-510   #18-516    18-516
RESMAP          #5-74      14-325    15-358    16-419    17-488
RESRG           #5-61      16-414    17-483
RETURN           8-132     10-206    11-228    12-246    13-284    14-326    15-359    16-420    17-489    18-523
                19-548     20-575    21-604    22-635    23-660
SAVMAP          #5-70      14-308    15-343    16-375    17-436
SAVRG           #5-61      16-385    17-444
STATE$          #5-62       9-148   #9-151    #9-155    #9-158    #9-161    #9-164    #9-167    #9-170    #9-173
                #9-177    #9-180    #9-184
TRAN$           #5-62     #9-149    #9-152    #9-153    #9-156    #9-159    #9-162    #9-165    #9-168    #9-171
                #9-174    #9-175    #9-178    #9-181    #9-182
VNPDBG          #5-63      13-257
```

```

                                  ;
                                  ; NETWORK NAME
                                  ;
 000352  016700  000000G          DTENET: MOV     .PSTCN,R0               ; GET SIZE OF SUPPLIED NAME
 000356  020027  000006                   CMP     R0,#DTNMMX              ; BIGGER THAN MAXIMUM?
 000362  101010                           BHI     101$                    ; BR IF YES, ERROR
 000364  016701  000000G                  MOV     .PSTPT,R1               ; POINT TO NAME
 000370  012702  000027'                  MOV     #DTNTNM,R2              ; POINT TO LOCAL COPY
 000374  112122                   10$:    MOVB    (R1)+,(R2)+             ; COPY NAME
 000376  005300                           DEC     R0                      ; MORE TO COPY?
 000400  003375                           BGT     10$                     ; BR IF YES
 000402  000403                           BR      DTEEND                  ; ELSE MERGE INTO END PROCESSING

                                  ;
                                  ; ERRORS
                                  ;
 000404                           101$:   EMSG$R  00                      ; ILLEGAL NETWORK NAME PARAMETER

```

```
**FILE**ID**VFPPSN







VV      VV   FFFFFFFFFF   PPPPPPPP     PPPPPPPP       SSSSSSSS   NN      NN
VV      VV   FFFFFFFFFF   PPPPPPPP     PPPPPPPP       SSSSSSSS   NN      NN
VV      VV   FF           PP      PP   PP      PP   SS           NN      NN
VV      VV   FF           PP      PP   PP      PP   SS           NN      NN
VV      VV   FF           PP      PP   PP      PP   SS           NNNN    NN
VV      VV   FF           PP      PP   PP      PP   SS           NNNN    NN
VV      VV   FFFFFFFF     PPPPPPPP     PPPPPPPP       SSSSSS     NN  NN  NN
VV      VV   FFFFFFFF     PPPPPPPP     PPPPPPPP       SSSSSS     NN  NN  NN
VV      VV   FF           PP           PP                   SS   NN    NNNN
VV      VV   FF           PP           PP                   SS   NN    NNNN
  VV  VV     FF           PP           PP                   SS   NN      NN       ....
  VV  VV     FF           PP           PP                   SS   NN      NN       ....
    VV       FF           PP           PP           SSSSSSSS     NN      NN       ....
    VV       FF           PP           PP           SSSSSSSS     NN      NN       ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
000716                              CALL    ALLPSV                  ; Attempt to allocate tables           ;[TD001]
000722  103412                      BCS     104$                    ; Branch if allocation error           ;[TD001]
                                                                                                           ;[TD001]
000724                      70$:    RETURN                                                                 ;[TD001]
                            ;                                                                              ;**-49
                            ; ERROR MESSAGES
                            ;
000726                      101$:   EMSG$R  D2                      ; ERROR - INVALID HIGH SUBADDRESS VALUE
000734                      102$:   EMSG$R  P5                      ; ERROR - HOME BLOCK ALLOCATION FAILURE
000742                      103$:   EMSG$R  P6                      ; ERROR - PORT TABLE ALLOCATION FAILURE
000750                      104$:   EMSG$R  PA                      ; Error - PSV mapping tables alocn.    ;[TD001]
                                                                                                           ;[TD001]
                                    .DSABL  LSB                                                            ;[TD001]
```

```
                                         .SBTTL  LOCAL DATA

                                 ;
                                 ; LOCAL DATA
                                 ;

    000000                               .PSECT  DATA,D

                                         .NLIST  BEX

                                 ;
                                 ; LOCAL DATA FOR PVC$DF MACRO
                                 ;
    000000                       PRTADD: .BLKW   1                ; ADDRESS OF ENTRY IN PORT TABLE
    000002                       PRTBIA: .BLKW   1                ; BIAS OF PORT TABLE
    000004                       PVCLCN: .BLKW   1                ; LOGICAL CHANNEL NUMBER
    000006                       PVCCTM: .BLKW   1                ; COUNTER TIMER
    000010                       PVCPSZ: .BLKW   1                ; MAXIMUM BLOCK SIZE
    000012                       PVCWSZ: .BLKW   1                ; WINDOW SIZE
    000014                       PVCOWN: .BLKW   1                ; OWNER PROCESS
    000016                       DLMPDV: .BLKW   1                ; ADDRESS OF DLM PDV
    000020                       DLMSLT: .BLKW   1                ; ADDRESS OF DLM SLT
    000022                       FLAG:   .BLKW   1                ; LOCAL FLAGS WORD
                000001                   PF.DLM = 1               ; PVC IS A DLM CIRCUIT
    000024                       PVCPRT: .BLKB   1                ; PORT NUMBER OF FREE ENTRY
    000025                       PVCFLG: .BLKB   1                ; PVC FLAGS BYTE
    000026                       PVCNML: .BLKB   1                ; Length of circuit identification          ;[TD001]
    000027                       PVCNAM: .BLKB   MAXID            ; Circuit identification                    ;[TD001]
    000047                       DLMCTL: .BLKB   1                ; DLM'S CONTROLLER NUMBER                   ;**-1
    000050          000          DLMUNT: .BYTE   0                ; DLM'S UNIT NUMBER (ALWAYS 0, MUST BE MTP)
    000051                       DLMSTA: .BLKB   1                ; DLM'S STATION NUMBER
    000052                       DLMIND: .BLKB   1                ; DLM'S PDV INDEX
                                         .EVEN                                                                ;[TD001]
                                 ;
                                 ; STATE VALUE FOR "DATA TRANSFER STATE" FOR XCB
                                 ; GLOBAL PATCHABLE JUST IN CASE ...
                                 ;
    000054      000003           SP$P4:: .WORD   3                ; VALUE FOR DATA TRANSFER STATE

```

```
VFGSVC     CREATED BY  MACRO  ON 29-JUN-85 AT 02:31      PAGE 4
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE             REFERENCES

T$LIFO     000013           #5-62
T$LIFS     000013           #5-62
T$LIN      000000           #5-62      15-574     15-576    *15-581
T$LIPS     000006           #5-62
T$LLD      000012           #5-62
T$LLDC     000045           #5-62
T$LLDL     000012           #5-62
T$LLDO     000012           #5-62
T$LLDS     000012           #5-62
T$LLEN     000046           #5-62      15-573     15-578
T$LOPR     000002           #5-62
T$LTCL     000024           #5-62
T$LTIM     000026           #5-62
T$LTPR     000014           #5-62
T$LTPS     000020           #5-62
T$NAPL     000004           #5-62
T$NFE      000000           #5-62
T$NLEN     000010           #5-62
T$NNUL     000002           #5-62
T$NOPL     000006           #5-62
T$NRNI     000042           #5-62
T$NRPL     000005           #5-62
T$NRUL     000007           #5-62
T$NVR      000001           #5-62
T$RPRI     000040           #5-62
T$SVC      000034           #5-62     *15-583    *15-588    *15-589
T$T5       000030           #5-62     *15-590
T$T6       000032           #5-62     *15-591
WNDSVC     000022 R         #6-127    *10-271    *11-326     13-459
Z.DAT    = ******   GX       15-568
Z.MAP    = ******   GX       15-565
Z.NAM    = ******   GX       14-515
$ALPHA   = 000022           #9-187
$ALPOL   = ******   GX       13-440
$ANY     = 000020           #9-187
$BFR     = ******   GX       13-448   *13-452    *13-453    *13-454    *13-455    *13-456    *13-457    *13-458    *13-459
                            *13-460    13-462    *13-467     13-472     15-569     15-570     15-571     15-574     15-576
                            *15-581   *15-582    *15-583    *15-588    *15-589    *15-590    *15-591
$BLANK   = 000006           #9-187
$CAT5    = ******   GX       10-267    13-424
$CDTB    = ******   GX       17-650
$CEACC   = ******   GX       13-445    13-474
$DIGIT   = 000024           #9-187
$DNUMB   = 000014           #9-187
$EOS     = 000012           #9-187
$ERR1T   = ******   GX        8-181
$ERR2I   = ******   GX       10-303
$ERR3D     000056 R         #7-140     11-317
$ERR3E     000102 R         #7-141     11-331
$ERR3F     000126 R         #7-142     11-343
$ERR3G     000152 R         #7-143     11-357
$ERR3H     000202 R         #7-144     12-369
```

```
    195                                             .SBTTL  X2P$DF ACTION ROUTINES
    196
    197                                     ;
    198                                     ; DEVICE NAME
    199                                     ;
    200                                             .ENABL  LSB
    201 000216  022767  000003  000000G X2.DEV: CMP     #3,.PSTCN               ; VALID DEVICE NAME?
    202 000224  103464                          BLO     101$                    ; BR IF NO
    203 000226  016700  000000G                 MOV     .PSTPT,R0               ; POINT TO START OF NAME
    204 000232                                  CALL    $CAT5                   ; CONVERT TO RAD50
    205 000236  010167  177536                  MOV     R1,X2DEV                ; STORE DEVICE NAME
    206 000242                                  RETURN
    207                                     ;
    208                                     ; DEVICE CONTROLLER NUMBER
    209                                     ;
    210 000244  005767  000000G         X2.CTL: TST     .PNUMH                  ; LEGAL CONTROLLER NUMBER?
    211 000250  001052                          BNE     101$                    ; BR IF NO
    212 000252  105767  000001G                 TSTB    .PNUMB+1                ; LEGAL?
    213 000256  001047                          BNE     101$                    ; BR IF NO
    214 000260  116767  000000G 177514          MOVB    .PNUMB,X2CTL            ; SAVE CONTROLLER NUMBER
    215 000266  105067  177511                  CLRB    X2UNT                   ; DEFAULT UNIT TO ZERO
    216 000272                                  RETURN
    217                                     ;
    218                                     ; DEVICE UNIT NUMBER
    219                                     ;
    220 000274  005767  000000G         X2.UNT: TST     .PNUMH                  ; LEGAL UNIT NUMBER?
    221 000300  001036                          BNE     101$                    ; BR IF NO
    222 000302  105767  000001G                 TSTB    .PNUMB+1                ; LEGAL?
    223 000306  001033                          BNE     101$                    ; BR IF NO
    224 000310  116767  000000G 177465          MOVB    .PNUMB,X2UNT            ; SAVE UNIT NUMBER
    225 000316  016700  000000G         CHKDEV: MOV     $SLTA,R0                ; GET SLT ADDRESS
    226 000322  116001  000002                  MOVB    L.DDM(R0),R1            ; GET DDM'S PDV INDEX
    227 000326  066701  000000G                 ADD     $PDVTA,R1               ; INDEX INTO PDV VECTOR TABLE
    228 000332  011101                          MOV     (R1),R1                 ; GET PDV INDEX
    229 000334  026167  000004  177436          CMP     Z.NAM(R1),X2DEV         ; IS THIS THE CORRECT DEVICE?
    230 000342  001010                          BNE     10$                     ; BR IF NO
    231 000344  126067  000012  177430          CMPB    L.CTL(R0),X2CTL         ; IS THIS THE CORRECT CONTROLLER?
    232 000352  001004                          BNE     10$                     ; BR IF NO
    233 000354  126067  000013  177421          CMPB    L.UNT(R0),X2UNT         ; IS THIS THE CORRECT UNIT NUMBER?
    234 000362  001404                          BEQ     20$                     ; BR IF YES - FOUND IT
    235 000364                          10$:    REJ$                            ; ELSE REJECT THE TRANSITION
    236 000374                          20$:    RETURN
    237
    238 000376                          101$:   EMSG$R  21                      ; ILLEGAL LINE-ID
    239                                         .DSABL  LSB
    240                                     ;
    241                                     ; MAXIMUM WINDOW SIZE
    242                                     ;
    243 000404  005767  000000G         X2.MWS: TST     .PNUMH                  ; LEGAL MAXIMUM WINDOW SIZE VALUE?
    244 000410  001013                          BNE     101$                    ; BR IF NO
    245 000412  016700  000000G                 MOV     .PNUMB,R0               ; GET MAXIMUM WINDOW SIZE
    246 000416  022700  000007                  CMP     #X2WNMX,R0              ; IS IT VALID?
    247 000422  103406                          BLO     101$                    ; BR IF NO
    248 000424  022700  000001                  CMP     #X2WNMN,R0              ; ...
    249 000430  101003                          BHI     101$                    ; BR IF NO
    250 000432  010067  000000G                 MOV     R0,$X2MWS               ; STORE MAXIMUM WINDOW SIZE
    251 000436                                  RETURN
```

```
     62                                       .SBTTL  CONSTANT DEFINITIONS
     63
     64                                 ;
     65                                 ; CONSTANT DEFINITIONS
     66                                 ;
     67          000040                       BLKSMN = 32.            ; MINIMUM BLOCK SIZE VALUE
     68          002000                       BLKSMX = 1024.          ; MAXIMUM BLOCK SIZE VALUE
     69          000177                       WNDSMX = 127.           ; MAXIMUM WINDOW SIZE VALUE
     70          000007                       X3TMCT = 7              ; NUMBER OF TIMER PARAMETERS IN X3P$DF
```

```
        .TITLE  VFIND - VNP FIND ENTITY ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       VNP - FIND ENTITY ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;


```

```
VFIND      CREATED BY  MACRO  ON 29-JUN-85 AT 02:33      PAGE 5
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE             REFERENCES

$CEACC   = ******  GX       7-98        8-140       9-188       10-226
$DECPT   = ******  GX       8-128       9-176       10-214
$FNADD     000330  RG      #9-174
$FNALI     000116  RG      #8-124
$FNNAM     000470  RG      #10-210
$FNOBJ     000002  RG      #7-89
$GETAD   = ******  GX       7-99        8-141       9-189       10-227
$GETRV   = ******  GX       8-128       9-176       10-214
$HMBUF   = ******  GX       8-127       8-131       9-175       9-179       10-213      10-217
$OBJHD   = ******  GX       7-90
$QUAL    = ******  GX       7-100       8-142       8-144
$RADDR   = ******  GX      *7-99       *8-128      *8-141      *9-176      *9-189      *10-214     *10-227
```

```
                                         ;
                                         ; CUG FLAGS BYTE
                                         ;
                                                 .ENABL  LSB
000226  005767  000000G                  CUFG:   TST     .PNUMH             ; LEGAL VALUE?
000232  001030                                   BNE     101$               ; BR IF NO
000234  105767  000001G                          TSTB    .PNUMB+1           ; LEGAL VALUE?
000240  001025                                   BNE     101$               ; BR IF NO
000242  116767  000000G 000032'                  MOVB    .PNUMB,CUGFLG      ; SAVE CUG FLAGS BYTE
000250                                   CUEND:                                                              ;[TD001]
000250  032767  000000G 000000G                  BIT     #PF.XDF,$PFLAG     ; Check for extended data format ;[TD001]
000256  001407                                   BEQ     10$                                                 ;[TD001]
                                         ;                                                                   ;[TD001]
                                         ; Process extended CUG format                                       ;[TD001]
                                         ;                                                                   ;[TD001]
000260                                           CALL    CKCUG0             ; Check for valid CUG name       ;[TD001]
000264  103416                                   BCS     102$               ; BR if error                    ;[TD001]
000266                                           CALL    ALCUG0             ; Allocate and fill CUG block    ;[TD001]
000272  103416                                   BCS     103$               ; BR if no store                 ;[TD001]
000274  000406                                   BR      70$                ; exit                           ;[TD001]
                                         ;                                                                   ;[TD001]
                                         ; Process old CUG format                                            ;[TD001]
                                         ;                                                                   ;[TD001]
000276                                   10$:    CALL    CKCUG1             ; Check for valid CUG name       ;[TD001]
000302  103407                                   BCS     102$               ; BR if error                    ;[TD001]
000304                                           CALL    ALCUG1             ; Allocate and fill CUG block    ;[TD001]
000310  103407                                   BCS     103$               ; BR if no store                 ;[TD001]
                                         ;                                                                   ;[TD001]
000312                                   70$:    RETURN                                                      ;[TD001]
                                         ;                                                                   ;**-37
                                         ; ERRORS
                                         ;
000314                                   101$:   EMSG$R  P4                 ; ILLEGAL FLAGS BYTE VALUE
000322                                   102$:   EMSG$R  N2                 ; ILLEGAL CUG NAME
000330                                   103$:   EMSG$R  M1                 ; CUG NAME BLOCK ALLOCATION FAILURE
                                                 .DSABL  LSB
```

```
                                        .TITLE  VFPDSA - INITIAL CONFIG FILE SCAN ACTION ROUTINES
                                        .IDENT  /V05.00/
                                        .ENABL  LC                                                              ;[TD001]
                                                                                                                ;**-1
                                ;
                                ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ; NTL - CONFIG FILE ACTION ROUTINES
                                ;
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  02-MAR-82
                                ;       DECNET-11M/S V3.0
                                ;       DECNET-11M-PLUS V1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;
```

```
VFPDSA      CREATED BY  MACRO  ON 29-JUN-85 AT 02:34      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE               REFERENCES

CERR    = ******  GX         7-77       7-78       7-79       7-80       7-81       7-82
CFGBF   = ******  GX         8-110
CFGSZ   = ******  GX         8-109
CFLIN   = ******  GX         7-77       7-78       7-79       7-80       7-81       7-82
CHKSB     000254 R          12-232     12-246    #12-260
COUNT   = ******  GX       *11-201     11-226
CUNMMX  = 000006           #6-70       13-277
DACUNM    000306 R         #13-275
DARDEN  = ******  GX        11-209
DARDND    000142 R         #11-221
DARDNO    000116 R         #11-209
DARDST    000060 R         #11-196
DARMLN  = ******  GX        11-196     11-222
DASSHI    000224 R         #12-246
DASSLO    000176 R         #12-232
DCMSK     000334 R         #14-291
DCVAL     000362 R         #14-304
DSACGL  = ******  GX        13-276
DSADF     000000 R           8-102
DSAKW     000000 RG          8-108     #9-128
DSARCT  = ******  GX       *11-226
DSARDP  = ******  GX        11-224
DSARDT  = ******  GX        11-197     11-202     11-223
DSASHI  = ******  GX       *12-235    *12-250
DSASLO  = ******  GX       *12-234     12-248
DSAST     000000 RG        #9-128
DSCDF     000130 R           8-105
DSCMCT  = ******  GX        14-292
DSCMSK  = ******  GX        14-291
DSCMX   = 000040           #6-69       14-316
DSCVAL  = ******  GX        14-304
DSCVCT  = ******  GX        14-305
FMT10   = ******  GX         7-77       7-78       7-79       7-80       7-81       7-82
FM.10   = 000000           #7-77      #7-78      #7-79      #7-80      #7-81      #7-82
I$$AS   = ******            14-334
NEXT    = ******  GX       *11-202     11-209
PCKBCD  = ******  GX        11-225
RTSPC   = ******  GX         7-77       7-78       7-79       7-80       7-81       7-82
R$$EIS  = ******            14-334
R$$11D  = ******            14-334
STORID  = ******  GX        13-278
STRHEX    000410 R          14-293     14-306    #14-314
STRNXT  = ******  GX        11-203     11-211
SUBAMX  = ******  GX        12-264
SYNERR  = ******  GX       *8-111     *8-114
S$$BAS  = ******             7-77       7-77       7-78       7-78       7-79       7-79       7-80       7-80       7-81
                             7-81       7-82       7-82

$ALPHA  = 000022           #9-128
$ANY    = 000020           #9-128
$BLANK  = 000006           #9-128
$DBG$   = ******             8-106
$DIGIT  = 000024           #9-128
```

```
                                   ;+
                                   ; *** DSNUSR - Check username
                                   ;
                                   ; INPUTS
                                   ;         STRBUF - Contains parsed string (minus delimiters etc)
                                   ;         STRLEN - Number of characters in string
                                   ;
                                   ; OUTPUTS
                                   ;         Username checked and stored
                                   ;         Error condition if invalid username
                                   ;-

000116                             DSNUSR:
000116   116701   000040'                    MOVB     STRLEN,R1             ; Get length of string
000122   020127   000020                     CMP      R1,#MXACUS            ; Check if username too long
000126   101011                              BHI      101$                  ; Branch if error
000130   110167   000000G                    MOVB     R1,RNUL               ; Move username length to data area
000134   012700   000041'                    MOV      #STRBUF,R0            ; Move username to data area
000140   012702   000000G                    MOV      #RNUSER,R2
000144                                       CALL     MOVE
000150                                       RETURN

000152                             101$:     EMSG$R   MC                    ; Illegal destination access data
```

```
     78                                                 .SBTTL  LOCAL SYMBOL DEFINITIONS
     79
     80                                            ;
     81                                            ; LOCAL SYMBOL DEFINITIONS
     82                                            ;
     83
     84            000040                               SPACE = 40              ; ASCII SPACE
     85            000006                               DSNAMX = 6              ; Max length of destination name [old format]   ;[TD001]
     86            000006                               DSTKMX = 6              ; Max length of object task name [old format]   ;[TD001]
     87            000006                               CUNMMX = 6              ; Max length of CUG name [old format]           ;[TD001]
     88            000020                               MAXID = 16.             ; Max length of net. mgmt. 'id-string'          ;[TD001]
     89            000006                               MXNOID = 6              ; Max length of node name                       ;[TD001]
     90                                                                                                                      ;[TD001]
```

```
    479 001432  116703  000000G                 MOVB    DSCMCT,R3               ; Get number of digits in call mask     ;[TD001]
    480 001436  006203                          ASR     R3                      ; Convert to number of bytes            ;[TD001]
    481 001440  005503                          ADC     R3                                                              ;[TD001]
    482 001442  110367  000030G                 MOVB    R3,$BFR+D$DATL          ; Move number of bytes to block         ;[TD001]
    483 001446                                  RESRG   <R1,R0>                 ; Restore address and length of block   ;[TD001]
    484 001452                                  PUTAD   R0,R1                   ; Write block back to image             ;[TD001]
    485 001466  010501                          MOV     R5,R1                   ; Get unmapped address of block         ;[TD001]
    486 001470                                  CALL    DSTINS                  ; Insert block into list                ;[TD001]
    487 001474  000241                          CLC                             ; Indicate block allocated              ;[TD001]
    488 001476                          100$:   RESMAP                          ; Restore previous mapping              ;[TD001]
    489 001502                                  RETURN                                                                  ;[TD001]
```

```
**FILE**ID**VFPDTE




VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD     TTTTTTTTTT   EEEEEEEEEE
VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD     TTTTTTTTTT   EEEEEEEEEE
VV      VV   FF           PP      PP   DD      DD       TT       EE
VV      VV   FF           PP      PP   DD      DD       TT       EE
VV      VV   FF           PP      PP   DD      DD       TT       EE
VV      VV   FF           PP      PP   DD      DD       TT       EE
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD       TT       EEEEEEEE
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD       TT       EEEEEEEE
VV      VV   FF           PP           DD      DD       TT       EE
VV      VV   FF           PP           DD      DD       TT       EE
  VV  VV     FF           PP           DD      DD       TT       EE               ....
  VV  VV     FF           PP           DD      DD       TT       EE               ....
    VV       FF           PP           DDDDDDDD         TT       EEEEEEEEEE       ....
    VV       FF           PP           DDDDDDDD         TT       EEEEEEEEEE       ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        ;
                                        ; END OF DTE$DF MACRO
                                        ;
000412                                  DTEEND: CALL    CHKDTE              ; CHECK FOR VALID DTE ADDRESS
000416  103575                                  BCS     101$                ; ERROR - DUPLICATE DTE ADDRESS
000420  020427  000020                          CMP     R4,#NUMDTE          ; MAXIMUM DTES ALREADY DEFINED?
000424  103402                                  BLO     30$                 ; No - another one is ok                ;[TD001]
000426  000167  000402                          JMP     104$                ; Yes- this one is too many             ;[TD001]
000432                                  30$:    CALL    CHKDEV              ; CHECK IF LINE EXISTS                  ;**-1
000436  103570                                  BCS     102$                ; BR IF LINE NOT IN SYSTEM

000440  016701  000000G                         MOV     HSHSZ,R1            ; GET NUMBER OF ENTRIES IN HASH
000444  006301                                  ASL     R1                  ; TIMES 4 TD GET
000446  006301                                  ASL     R1                  ;  SIZE IN BYTES
000450  062701  000122                          ADD     #L$LEN,R1           ; PLUS DTE SIZE GETS TOTAL SIZE
000454  032767  000000G 000000G                 BIT     #PF.PSV,$PFLAG      ; Check if gateway extension present    ;[TD001]
000462  001402                                  BEQ     35$                 ; Branch if not needed                  ;[TD001]
000464  062701  000012                          ADD     #L$GLEN-L$LEN,R1    ; Add on size of extension              ;[TD001]
                                                                                                                    ;[TD001]
000470                                  35$:    CALL    $SAVAL              ; Save all registers                    ;[TD001]
000474                                          CALL    $ALPOL              ; ALLOCATE BLOCK FROM EXTENDED POOL     ;**-2
000500  103552                                  BCS     103$                ; BR IF UNSUCCESSFUL ALLOCATION

000502                                          SAVMAP                      ; SAVE CURRENT MAPPING
000506  010067  000000G                         MOV     R0,DTEDES           ; SAVE DTE DESCRIPTOR ADDRESS
000512                                          CEACC$  R0,R3               ; CONVERT TO MAPPED ADDRESS
000522  012702  000000G                         MOV     #$BFR,R2            ; POINT TO DTE DESCRIPTOR BLOCK
000526  105022                          40$:    CLRB    (R2)+               ; INITIALIZE BLOCK
000530  005301                                  DEC     R1                  ; MORE TO INITIALIZE?
000532  003375                                  BGT     40$                 ; BR IF YES

000534  012701  000000G                         MOV     #$BFR,R1            ; GET ADDRESS OF DTE DESCRIPTOR BLOCK
000540  062701  000020                          ADD     #L$DTEA,R1          ; POINT TO DTE ADDRESS FIELD
000544  012702  000010                          MOV     #<DTEALN+1>/2,R2    ; GET LENGTH OF ADDRESSS FIELD
000550  012705  000017'                         MOV     #DTEPCK,R5          ; GET ADDRESS OF SPECIFIED ADDRESS
000554  112521                          45$:    MOVB    (R5)+,(R1)+         ; STORE DTE ADDRESS
000556  005302                                  DEC     R2                  ; MORE TO STORE?
000560  003375                                  BGT     45$                 ; BR IF YES
000562  116767  000000G 000017G                 MOVB    COUNT,L$DTEL+$BFR   ; STORE DTE ADDRESS LENGTH

000570  012701  000000G                         MOV     #$BFR,R1            ; GET ADDRESS OF BLOCK
000574  062701  000114                          ADD     #L$NETW,R1          ; POINT TO NETWORK NAME FIELD
000600  012702  000006                          MOV     #DTNMMX,R2          ; GET LENGTH OF NAME
000604  012705  000027'                         MOV     #DTNTNM,R5          ; POINT TO SPECIFIED NAME
000610  112521                          50$:    MOVB    (R5)+,(R1)+         ; STORE NETWORK NAME
000612  005302                                  DEC     R2                  ; MORE TO STORE?
000614  003375                                  BGT     50$                 ; BR IF YES

000616  110467  000004G                         MOVB    R4,L$SLN+$BFR       ; STORE SYSTEM LINE NUMBER
000622  006304                                  ASL     R4                  ; GET SYSTEM LINE NUMBER X 2
000624  016702  000000G                         MOV     $LLCTA,R2           ; POINT TO REVERSE MAPPING TABLE
000630  060204                                  ADD     R2,R4               ; INDEX INTO REVERSE MAPPING TABLE
000632  111467  000016G                         MOVB    (R4),L$LLCH+$BFR    ; STORE CHANNEL NUMBER
000636  016767  000000G 000040G                 MOV     CTIM,L$CTIM+$BFR    ; STORE COUNTER TIMER
000644  005067  000044G                         CLR     L$TCLZ+$BFR         ; STORE TIME COUNTERS LAST ZEROED
000650  112767  000001  000003G                 MOVB    #LN$OFF,L$NMST+$BFR ; STORE ASSUMED OFF STATE
000656  032767  000000G 000000G                 BIT     #DE.ON,DTEFLG       ; SET DTE STATE ON?
```

```
VFPPSN - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:37
Table of contents
```

```
     5-   56    MACRO DEFINITIONS
     6-   82    LOCAL SYMBOLS
     8-  103    ERROR MESSAGES
     9-  119    LOOK FOR PSN$DF MACRO
    10-  150    TPARS STATE TABLES
    11-  200    PSN$DF ACTION ROUTINES
    14-  369    ALLPRT - Allocate port table                                  ;[TD001]
    15-  407    ALLPSV - Allocate PSV mapping tables                          ;[TD001]
```

```
                                      .SBTTL  ALLPRT - Allocate port table                                          ;[TD001]
                              ;+                                                                                    ;[TD001]
                              ; *** ALLPRT - Allocate port table                                                    ;[TD001]
                              ;                                                                                     ;[TD001]
                              ; INPUTS                                                                              ;[TD001]
                              ;       $BFR - used as work area                                                      ;[TD001]
                              ;       PORTNO - number of entries in table                                           ;[TD001]
                              ;       $PSIPT - pointer to PSI home block                                            ;[TD001]
                              ;                                                                                     ;[TD001]
                              ; OUTPUTS                                                                             ;[TD001]
                              ;       Carry clear:    Port table allocated                                          ;[TD001]
                              ;       Carry set:      Allocation failed                                             ;[TD001]
                              ;-                                                                                    ;[TD001]
                                                                                                                    ;[TD001]
000756                        ALLPRT:                                                                               ;[TD001]
000756  016701  000000G               MOV     PORTNO,R1               ; Find number of ports required               ;[TD001]
000762  006301                        ASL     R1                      ; Calculate bytes required                    ;[TD001]
000764                                CALL    $ALPOL                  ; Attempt to allocate                         ;[TD001]
000770  103447                        BCS     99$                     ; Error if insufficient store                 ;[TD001]
000772  010003                        MOV     R0,R3                   ; Save port table address                     ;[TD001]
000774                                SAVMAP                          ; Save current mapping                        ;[TD001]
001000                                CEACC$  R0                      ; Map to port table                           ;[TD001]
001010  012702  000000G               MOV     #$BFR,R2                ; Address port table                          ;[TD001]
001014  016701  000000G               MOV     PORTNO,R1               ; Get number of words in table                ;[TD001]
001020  005022                10$:    CLR     (R2)+                   ; Clear port table                            ;[TD001]
001022  005301                        DEC     R1                                                                    ;[TD001]
001024  003375                        BGT     10$                                                                   ;[TD001]
001026  016701  000000G               MOV     PORTNO,R1               ; Calculate size of port table                ;[TD001]
001032  006301                        ASL     R1                                                                    ;[TD001]
001034                                PUTAD   R0,R1                   ; Write updated port table                    ;[TD001]
001050                                RESMAP                          ; Restore mapping                             ;[TD001]
001054  016700  000000G               MOV     $PSIPT,R0               ; Get address of PSI mapping table            ;[TD001]
001060  062700  000020                ADD     #H$PTB,R0               ; Get address of Port Table pointer           ;[TD001]
001064  010367  000000G               MOV     R3,$BFR                 ; Store address of Port Table                 ;[TD001]
001070                                PUTRC   R0,#2                   ; Write Port Table address to image           ;[TD001]
001106  000241                        CLC                             ; Indicate succesful allocation               ;[TD001]
001110                        99$:    RETURN                                                                        ;[TD001]
```

```
                        .SBTTL  ERROR MESSAGES

        000056          NTLER$  ,N0,10,CERR,RTSPC,CFLIN,<circuit identification>              ;**-2
        000116          NTLER$  ,N1,10,CERR,RTSPC,CFLIN,<process owner name>
        000152          NTLER$  ,N2,10,CERR,RTSPC,CFLIN,<flags byte value>
        000204          NTLER$  ,N3,8,CERR,RTSPC,CFLIN,<Resource allocation failure>

                        .EVEN                                                                 ;**-1
                        .LIST   BEX
        000000          .PSECT
```

```
VFGSVL      CREATED BY  MACRO  ON 29-JUN-85 AT 02:31      PAGE 5
SYMBOL CROSS REFERENCE                                     CREF    04.00

SYMBOL  VALUE                REFERENCES

$ERR3I    000230 R          #7-145      12-404
$ERR3J    000254 R          #7-146      13-489
$ERR3K    000272 R          #7-147      13-491
$ERR3M    000344 R          #7-148      12-380
$ERR3Z    000400 R          #7-149      13-490
$EXIT   = 000000            #9-187
$FAIL   = 177777            #9-187
$GETAD  = ******    GX       13-475     15-573
$GETRV  = ******    GX       13-470     15-568
$GPRM   = ******              9-187
$LAMDA  = 000000            #9-187
$LLCTA  = ******    GX       15-550
$NUMBR  = 000002            #9-187
$PDVNM  = ******    GX       14-511
$PDVTA  = ******    GX       14-510     14-522      15-562
$PSIPT  = ******    GX       13-434
$PUTAD  = ******    GX       13-468     15-592
$QSVC     000000 RG         #8-167
$RADDR  = ******    GX      *13-468    *13-470     *13-475     *15-568     *15-573
$RAD50  = 000016            #9-187
$RBLCK  = ******    GX       13-443     13-447     *13-483      15-567     *15-569     *15-593
$RONLY  = ******              9-187      9-187       9-187
$SAVAL  = ******    GX       13-437
$SAVRG  = ******    GX       12-409
$SLTA   = ******    GX       10-282     13-429
$SLTMA  = ******    GX       15-545     15-549
$STRNG  = 000004            #9-187
$SUBXP  = 000010            #9-187
$TRIB   = ******    GX       10-296
$XLINK  = ******    GX       13-479
$$HTIM  = ******    GX       13-432
$$LTIM  = ******    GX       13-433
$$$FLG  = 177777            #9-187
$$$KEY  = 177777            #9-187
.PCHAR  = ******    GX       17-648
.PNUMB  = ******    GX       10-279     10-281      10-283      10-293      10-295      10-296      11-312      11-324      11-326
                             11-338     11-350      11-352      12-364
.PNUMH  = ******    GX       10-277     10-291      11-310      11-322      11-336      11-348      12-362
.PSTCN  = ******    GX       10-263     13-419
.PSTPT  = ******    GX       10-265     13-422
.TPARS  = ******    GX       8-173
```

```
000440                                  101$:   EMSG$R  2F              ; ILLEGAL MAXIMUM WINDOW SIZE VALUE
                                        ;
                                        ; MAXIMUM BLOCK SIZE
                                        ;
000446  005767  000000G                 X2.MBS: TST     .PNUMH          ; LEGAL MAXIMUM BLOCK SIZE VALUE?
000452  001005                                  BNE     101$            ; BR IF NO
000454  016700  000000G                         MOV     .PNUMB,R0       ; GET MAXIMUM BLOCK SIZE
000460  010067  000000G                         MOV     R0,$X2MBS       ; STORE MAXIMUM BLOCK SIZE
000464                                          RETURN
000466                                  101$:   EMSG$R  2C              ; ILLEGAL MAXIMUM BLOCK SIZE
                                        ;
                                        ; MAXIMUM RETRANSMIT COUNT
                                        ;
000474  005767  000000G                 X2.MRT: TST     .PNUMH          ; LEGAL RETRANSMIT COUNT?
000500  001007                                  BNE     101$            ; BR IF NO
000502  105767  000001G                         TSTB    .PNUMB+1        ; ...
000506  001004                                  BNE     101$            ; BR IF NO
000510  016767  000000G 000000G                 MOV     .PNUMB,$X2MRT   ; STORE MAXIMUM RETRANSMIT COUNT
000516                                          RETURN
000520                                  101$:   EMSG$R  2J              ; ILLEGAL MAXIMUM RETRANSMIT COUNT
                                        ;
                                        ; RETRANSMIT TIMER
                                        ;
000526  005767  000000G                 X2.RTR: TST     .PNUMH          ; LEGAL RETRANSMIT TIMER?
000532  001011                                  BNE     101$            ; BR IF NO
000534  016700  000000G                         MOV     .PNUMB,R0       ; GET RETRANSMIT TIMER
000540  002406                                  BLT     101$            ; BR IF ILLEGAL VALUE
000542  022700  001130                          CMP     #X2RTMX,R0      ; LEGAL RETRANSMIT TIMER VALUE?
000546  103403                                  BLO     101$            ; BR IF ILLEGAL VALUE
000550  010067  000000G                         MOV     R0,$X2RET       ; STORE RETRANSMIT TIMER
000554                                          RETURN
000556                                  101$:   EMSG$R  2K              ; ILLEGAL RETRANSMIT TIMER
                                        ;
                                        ; HOLDBACK TIMER
                                        ;
000564  005767  000000G                 X2.HLD: TST     .PNUMH          ; LEGAL HOLDBACK TIMER?
000570  001011                                  BNE     101$            ; BR IF NO
000572  016700  000000G                         MOV     .PNUMB,R0       ; GET HOLDBACK TIMER VALUE
000576  002406                                  BLT     101$            ; BR IF ILLEGAL VALUE
000600  022700  001130                          CMP     #X2HLMX,R0      ; LEGAL HOLDBACK TIMER VALUE?
000604  103403                                  BLO     101$            ; BR IF NO
000606  010067  000000G                         MOV     R0,$X2HLD       ; STORE HOLDBACK TIMER VALUE
000612                                          RETURN
000614                                  101$:   EMSG$R  2L              ; ILLEGAL HOLDBACK TIMER VALUE

        000001                                  .END
```

```
     72                                      .SBTTL  LOCAL DATA
     73
     74 000000                               .PSECT  DATA,D
     75                                 ;
     76                                 ; LOCAL DATA
     77                                 ;
     78 000000                          NEXT:   .BLKW   1               ; ADDRESS OF NEXT TIMER
     79 000002                          X3DEV:  .BLKW   1               ; RAD50 DEVICE NAME
     80 000004                          X3CTL:  .BLKW   1               ; CONTROLLER NUMBER
     81 000006                          X3UNT:  .BLKW   1               ; UNIT NUMBER
     82 000010                          TIMRCT: .BLKB   1               ; COUNT OF NUMBER OF TIMER PARAMETERS SEEN
     83                                         .EVEN
     54                                         .SBTTL  MACRO DEFINTIONS
     55
     56                                 ;****
     57                                 ; MACRO CALLS
     58                                 ;****
     59
     60                                         .MCALL  OBJDF$,RNBDF$,GETAD,ANBDF$,VNPDF$,CEACC$,GETRV,DHBDF$
     61                                         .MCALL  SWTCHI,SWTCHO
     62
     63 000000                                  OBJDF$                          ; DEFINE OBJECT BLOCK OFFSETS
     64 000000                                  RNBDF$                          ; DEFINE REMOTE BLOCK OFFSETS
     65 000000                                  ANBDF$                          ; DEFINE ALIAS BLOCK OFFSETS
     66 000000                                  VNPDF$                          ; DEFINE REQUEST BLOCK OFFSETS
     67 000000                                  DHBDF$                          ; DEFINE DECNET HOME BLOCK OFFSETS
     68
```

```
VFIND      CREATED BY  MACRO  ON 29-JUN-85 AT 02:33     PAGE 6
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

ANBDF$           #5-60       5-65
CALL              7-98       7-99        8-128       8-140       8-141       9-176       9-188       9-189       10-214      10-226
                  10-227
CEACC$           #5-60       7-98        8-140       9-188       10-226
DHBDF$           #5-60       5-67
GETAD            #5-60       7-99        8-141       9-189       10-227
GETRV            #5-60       8-128       9-176       10-214
OBJDF$           #5-60       5-63
RAND             #7-99       7-99       #8-128       8-128      #8-141       8-141      #9-176       9-176      #9-189       9-189
                 #10-214     10-214     #10-227      10-227
RETURN            7-107      8-159       9-195       10-239
RNBDF$           #5-60       5-64
SWTCHI           #5-61       8-127       8-129       9-175       9-177       10-213      10-215
SWTCHO           #5-61
VNPDF$           #5-60       5-66
```

```
                                                 .SBTTL  CKCUGO - Check for unique cug name (new format)          ;[TD001]
                                                                                                                   ;**-1
                                       ;+
                                       ;
                                       ; *** CKCUGO - Check for unique cug name (new format)                      ;[TD001]
                                       ;                                                                          ;[TD001]
                                       ; INPUTS:                                                                  ;[TD001]
                                       ;        CUGNAM - Specified cug name                                       ;[TD001]
                                       ;        CUGNML - Length of CUG name specified                             ;[TD001]
                                       ;                                                                          ;[TD001]
                                       ; OUTPUTS:                                                                 ;[TD001]
                                       ;        Carry clear - specified cug name is unique                        ;[TD001]
                                       ;        Carry set - specified cug name is not unique                      ;[TD001]
                                       ;                                                                          ;[TD001]
                                       ;-                                                                         ;[TD001]
000336                                 CKCUGO: CALL    $SAVAL                  ; Save all registers              ;[TD001]
000342  016700  000000G                        MOV     $PSIPT,R0               ; Get address of PSI home block   ;[TD001]
000346  062700  000010                         ADD     #H$CUG,R0               ; Point to CUG listhead           ;[TD001]
000352                                         GETRV   R0,#2                   ; Get listhead                    ;[TD001]
000370                                         SAVMAP                          ; Save current mapping            ;[TD001]
000374  016700  000000G                10$:    MOV     $BFR,R0                 ; Get address of next block in list ;[TD001]
000400  000241                                 CLC                             ; Assume success                  ;[TD001]
000402  001435                                 BEQ     25$                     ; Branch if end of list (name unique) ;[TD001]
000404                                         CEACC$  R0                      ; Convert to mapped address       ;[TD001]
000414                                         GETAD   R0,#G$LEN+MAXID         ; Read in enough to contain CUG name ;[TD001]
000432  026767  000012G 000000G                CMP     G$DTE+$BFR,DTEDES       ; Same DTE as new block?          ;[TD001]
000440  001355                                 BNE     10$                     ; No, so don't check name         ;[TD001]
000442  116702  000000'                        MOVB    CUGNML,R2               ; Get length of CUG name          ;[TD001]
000446  120267  000015G                        CMPB    R2,$BFR+G$NML           ; Compare with length in CUG block ;[TD001]
000452  001350                                 BNE     10$                     ; NE, not same CUG                ;[TD001]
000454  012701  000001'                        MOV     #CUGNAM,R1              ; Point to CUG name               ;[TD001]
000460  012703  000016G                        MOV     #$BFR+G$GNAM,R3         ; Point to CUG name block         ;[TD001]
000464  122123                         20$:    CMPB    (R1)+,(R3)+             ; Is the name unique?             ;[TD001]
000466  001342                                 BNE     10$                     ; Branch if yes                   ;[TD001]
000470  005302                                 DEC     R2                      ; More to check?                  ;[TD001]
000472  003374                                 BGT     20$                     ; Branch if yes                   ;[TD001]
000474  000261                                 SEC                             ; Else error - set C-bit          ;[TD001]
000476                                 25$:    RESMAP                          ; Restore previous mapping        ;[TD001]
000502                                 30$:    RETURN                                                            ;[TD001]
```

```
                        .SBTTL  MACRO DEFINITIONS

                ;
                ; LIBRARY MACROS
                ;
                        .MCALL  EMSG$R,NTLER$,SAVRG,RESRG,ASL$,STATE$,TRAN$,ISTAT$
                        .MCALL  VNPDBG                                                  ;[TD001]
```

```
VFPDSA      CREATED BY  MACRO  ON 29-JUN-85 AT 02:34      PAGE 2
SYMBOL  CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE                REFERENCES

$DNUMB   = 000014             #9-128
$EOS     = 000012             #9-128
$ERRM0     000106 R           #7-80          12-255
$ERRM2     000136 R           #7-81          14-312
$ERRM3     000166 R           #7-82          14-299
$ERRN2     000000 R           #7-77          13-284
$ERRN8     000022 R           #7-78          11-216
$ERRN9     000056 R           #7-79          12-240
$ERR1T   = ******   GX         8-121
$EXIT    = 000000             #9-128
$FAIL    = 177777             #9-128
$GPRM    = ******              9-128
$LAMDA   = 000000             #9-128
$NUMBR   = 000002             #9-128
$QDSA      000000 RG          #8-102
$QDSC      000006 RG          #8-104
$RAD50   = 000016             #9-128
$RONLY   = ******              9-128         9-128         9-128
$SAVRG   = ******   GX         11-221
$STRNG   = 000004             #9-128
$SUBXP   = 000010             #9-128
$$$FLG   = 177777             #9-128
$$$KEY   = 177777             #9-128
.PNUMB   = ******   GX         12-262
.PNUMH   = ******   GX         12-260
.PSTCN   = ******   GX         14-315
.PSTPT   = ******   GX         14-319
.TPARS   = ******   GX         8-112
```

```
                                             ;+
                                             ; *** DSNPAS - Check password
                                             ;
                                             ; INPUTS
                                             ;       STRBUF - Contains parsed string (minus delimiters etc)
                                             ;       STRLEN - Number of characters in string
                                             ;
                                             ; OUTPUTS
                                             ;       Password checked and stored
                                             ;       Error condition if invalid password
                                             ;-
000160                                       DSNPAS:
000160   116701  000040'                             MOVB    STRLEN,R1               ; Get length of string
000164   020127  000010                              CMP     R1,#MXACPA              ; Check if password too long
000170   101011                                      BHI     101$                    ; Branch if error
000172   110167  000000G                             MOVB    R1,RNPL                 ; Move password length to data area
000176   012700  000041'                             MOV     #STRBUF,R0              ; Move password to data area
000202   012702  000000G                             MOV     #RNPASS,R2
000206                                               CALL    MOVE
000212                                               RETURN

000214                                       101$:   EMSG$R  MC                      ; Illegal destination access data
```

```
                 .SBTTL  ERROR MESSAGES                                                  ;[TD001]
        000000   .PSECT  DATA,D                                                          ;[TD001]
                                                                                          ;**-18
        000000   NTLER$  ,N4,10,CERR,RTSPC,CFLIN,<destination name>
        000032   NTLER$  ,N5,10,CERR,RTSPC,CFLIN,<priority number>
        000062   NTLER$  ,N6,10,CERR,RTSPC,CFLIN,<object number>
        000110   NTLER$  ,N7,10,CERR,RTSPC,CFLIN,<task name>
        000132   NTLER$  ,6M,10,CERR,RTSPC,CFLIN,<call data mask/value combination>
        000204   NTLER$  ,M7,10,CERR,RTSPC,CFLIN,<destination name>
        000236   NTLER$  ,M9,8,CERR,RTSPC,CFLIN,<Destination block allocation failure>

                 .LIST   BEX                                                             ;**-1
                 .EVEN
        000000   .PSECT
```

```
                                                 .SBTTL  DSTINS - INSERT DESTINATION DESCRIPTOR BLOCK INTO LIST

                                        ;+
                                        ;
                                        ; DSTINS - INSERT DESTINATION DESCRIPTOR BLOCK INTO LIST
                                        ;
                                        ; INPUTS:
                                        ;       R1 - UNMAPPED ADDRESS OF BLOCK TO LINK
                                        ;       DSTPRI - SPECIFIED PRIORITY
                                        ;
                                        ; OUTPUTS:
                                        ;       BLOCK IS LINKED INTO LIST BY PRIORITY
                                        ;       R2,R3,R4 DESTROYED
                                        ;-
001504  016702  000000G                 DSTINS: MOV     $PSIPT,R2               ; GET ADDRESS OF PSI HOME BLOCK
001510  062702  000012                          ADD     #H$DST,R2               ; POINT TO X25 LISTHEAD
001514  132767  000000G 000000G                 BITB    #DS.X29,DSFLG           ; IS THIS AN X29 DESCRIPTOR BLOCK?
001522  001402                                  BEQ     10$                     ; NO, X25 IT IS
001524  062702  000002                          ADD     #H$D29-H$DST,R2         ; ELSE, POINT TO X29 LISTHEAD
001530                                  10$:    GETRV   R2,#2                   ; READ IN LISTHEAD

001546  010204                          20$:    MOV     R2,R4                   ; SAVE UNMAPPED ADDRESS OF PREVIOUS
001550  016702  000000G                         MOV     $BFR,R2                 ; GET UNMAPPED ADDRESS OF NEXT
001554  001417                                  BEQ     40$                     ; BR IF END OF LIST - LINK HERE
001556                                          CEACC$  R2,R3                   ; GET MAPPED ADDRESS
001566                                          GETAD   R3,#D$PRI+1             ; Read in dest. block link and priority ;[TD001]
001604  126767  000000G 000002G                 CMPB    DSTPRI,D$PRI+$BFR       ; LINK BLOCK BEFORE THIS ONE?           ;**-1
001612  101755                                  BLOS    20$                     ; BR IF NO

001614  010146                          40$:    MOV     R1,-(SP)                ; GET UNMAPPED ADDR OF BLOCK TO LINK
001616  010446                                  MOV     R4,-(SP)                ; GET ADDRESS OF PREVIOUS BLOCK
001620                                          CALL    $XLINK                  ; LINK BLOCK INTO LIST
001624                                          RETURN                          ;
```

```
VFPDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:36
Table of contents
```

```
     5-   56    MACRO DEFINITIONS
     6-   81    LOCAL SYMBOL DEFINITIONS
     7-  101    LOCAL DATA
     8-  123    ERROR MESSAGES
     9-  135    LOOK FOR DTE$DF MACRO
    10-  166    TPARS STATE TABLES
    11-  218    LINE-ID SUBEXPRESSION
    12-  241    DTE$DF ACTION ROUTINES
    16-  431    CHKDTE - CHECK FOR VALID DTE ADDRESS
    17-  471    CHKDEV - CHECK VALIDITY OF LINE-ID
    18-  523    BIASX - SET APR6 BIAS CORRECTLY
```

```
000664  001403                           BEQ     60$                     ; BR IF NO- OFF IT IS
000666  112767  000000  000003G          MOVB    #LN$ON,L$NMST+$BFR      ; STORE ON STATE
000674  116767  000003G 000002G 60$:     MOVB    L$NMST+$BFR,L$OMST+$BFR ; SET PREVIOUS STATE AS WELL
000702  010300                           MOV     R3,R0                   ; COPY DTE ADDRESS
000704  062700  000122                   ADD     #L$LEN,R0               ; OFFSET TO START OF HASH TABLE
000710  032767  000000G 000000G          BIT     #PF.PSV,$PFLAG          ; Check if gateway extension present     ;[TD001]
000716  001402                           BEQ     65$                     ; Branch if not needed                   ;[TD001]
000720  062700  000012                   ADD     #L$GLEN-L$LEN,R0        ; Add on size of extension               ;[TD001]
000724  010067  000000G         65$:     MOV     R0,HSHADD               ; Save hash table address                ;[TD001]
000730                                   CALL    BIASX                   ; SET CORRECT BIAS ON ADDRESS            ;**-1
000734  010067  000030G                  MOV     R0,L$CHTB+$BFR          ; STORE HASH TABLE ADDRESS
000740  016767  000000G 000032G          MOV     HSHSZ,L$CTEN+$BFR       ; SAVE NUMBER OF ENTRIES
000746                                   PUTAD   R3,R$R1+2(SP)           ; WRITE UPDATED DTE DESCRIPTOR
000764  016700  000000G                  MOV     $PSIPT,R0               ; GET ADDRESS OF PSI HOME BLOCK
000770  062700  000002                   ADD     #H$LDTE,R0              ; POINT TO DTE DESCRIPTOR LISTHEAD
000774  016705  000000G                  MOV     DTEDES,R5               ; GET UNMAPPED ADDR OF BLOCK TO LINK
001000                                   CALL    LNKEND                  ; LINK BLOCK AT END OF LIST
001004                                   RESMAP                          ; RESTORE PREVIOUS MAPPING
001010                                   RETURN
                                ;
                                ; ERRORS
                                ;
001012                          101$:    EMSG$R  03                      ; ILLEGAL DTE ADDRESS
001020                          102$:    EMSG$R  07                      ; LINE NOT IN SYSTEM
001026                          103$:    EMSG$R  0P                      ; ALLOCATION FAILURE
001034                          104$:    EMSG$R  0Q                      ; TOO MANY DTES DEFINED
```

```
     .TITLE  VFPPSN - INITIAL CONFIG FILE SCAN ACTION ROUTINES
     .IDENT  /V05.00/                                                              ;[TD001]
     .ENABL  LC                                                                    ;**-1

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-82
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                         .SBTTL  ALLPSV - Allocate PSV mapping tables                      ;[TD001]
                                 ;+                                                                        ;[TD001]
                                 ; *** ALLPSV - Allocate PSV mapping tables                                ;[TD001]
                                 ;                                                                         ;[TD001]
                                 ; INPUTS                                                                  ;[TD001]
                                 ;       $BFR - used as work area                                          ;[TD001]
                                 ;       $PSIPT - pointer to PSI homeblock                                 ;[TD001]
                                 ;                                                                         ;[TD001]
                                 ; OUTPUTS                                                                 ;[TD001]
                                 ;       Carry clear: PSV mapping tables allocated                         ;[TD001]
                                 ;       Carry set:   Insufficient store                                   ;[TD001]
                                 ;       Registers destroyed                                               ;[TD001]
                                 ;-                                                                        ;[TD001]
001112                           ALLPSV:                                                                   ;[TD001]
001112  012702  000000G                  MOV     #$BFR,R2                ; Clear table-sized chunk of $BFR ;[TD001]
001116  012701  000400                   MOV     #PSVTSZ,R1                                                ;[TD001]
001122  005022                   10$:    CLR     (R2)+                                                     ;[TD001]
001124  005301                           DEC     R1                                                        ;[TD001]
001126  003375                           BGT     10$                                                       ;[TD001]
001130  012701  002000                   MOV     #<4*PSVTSZ>,R1          ; Length of both tables           ;[TD001]
001134                                   CALL    $ALPOL                  ; Attempt to allocate both tables ;[TD001]
001140  103456                           BCS     99$                     ; Error if no store               ;[TD001]
001142  010004                           MOV     R0,R4                   ; Save unmapped allocated address ;[TD001]
001144                                   SAVMAP                          ; Save current mapping            ;[TD001]
001150                                   CEACC$  R0                      ; Map to allocation               ;[TD001]
001160                                   PUTAD   R0,#<2*PSVTSZ>          ; Write first mapping table       ;[TD001]
001176  010400                           MOV     R4,R0                   ; Restore unmapped allocated address ;[TD001]
001200  062700  001000                   ADD     #<2*PSVTSZ>,R0          ; Address of second half of allocation ;[TD001]
001204  010005                           MOV     R0,R5                   ; Save unmapped address of second table ;[TD001]
001206                                   CEACC$  R0                      ; Map to second part of allocation ;[TD001]
001216                                   PUTAD   R0,#<2*PSVTSZ>          ; Write second mapping table      ;[TD001]
001234                                   RESMAP                          ; Restore mapping                 ;[TD001]
001240                                   GETRV   $PSIPT,#H$GLEN          ; Read PSI homeblock              ;[TD001]
001260  010467  000044G                  MOV     R4,$BFR+H$GLT           ; Store address of mapping tables ;[TD001]
001264  010567  000046G                  MOV     R5,$BFR+H$GPT                                             ;[TD001]
001270                                   PUTRC                           ; Write out updated home block    ;[TD001]
001274  000241                           CLC                             ; Successful allocation           ;[TD001]
001276                           99$:    RETURN                                                            ;[TD001]
                                                                                                           ;[TD001]
        000001                           .END                                                              ;**-1
```

```
                                        .SBTTL  LOOK FOR PVC$DF MACRO

                                ;+
                                ; $QPVC - LOOK FOR PVC$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3,R4,R5=DESTROYED
                                ;-
                                                                                                        ;[TD001]
000000                          $QPVC:: VNPDBG  VFPPVC          ; VNP debugging break point            ;[TD001]
000000  005067  000022'                 CLR     FLAG            ; CLEAR LOCAL FLAGS WORD               ;[TD001]
000004  012705  000000'                 MOV     #PVCDF,R5       ; STATE TABLE ADDRESS                  ;**-1
000010  005001                          CLR     R1              ; FULL KEYWORD MATCH LENGTH
000012  012702  000000'                 MOV     #PCVKW,R2       ; KEYWORD TABLE ADDRESS
000016  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
000022  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000026  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000032                                  CALL    .TPARS          ; GO DO THE PARSE
000036  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000040  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000044  001401                          BEQ     101$            ; IF EQ, YES
000046                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000050                          101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
VFGSVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:31      PAGE 6

MACRO CROSS REFERENCE                                     CREF    04.00

MACRO NAME       REFERENCES

ASL$                 #5-60    16-623
BIAS                #5-102
CALL                 8-173    10-267    12-391    12-399    12-413    13-424    13-427    13-437    13-440    13-445
                    13-468    13-470    13-474    13-475    13-479    13-482    15-568    15-573    15-592    17-650
CEACC$               #5-60    13-445    13-474
CTRDF$               #5-59      5-62
DBGTP$              #9-187    #9-214    #9-221    #9-246    #9-250
EMSG$R               #5-59     8-181    10-303    11-317    11-331    11-343    11-357    12-369    12-380    12-404
                    13-489    13-490    13-491
GETAD                #5-60    13-475    15-573
GETRV                #5-60    13-470    15-568
ISTAT$               #5-59     9-187
MAP                  #5-96    15-569
MTRAN$              #9-187
NTLER$               #5-59     7-140     7-141     7-142     7-143     7-144     7-145     7-146     7-147     7-148
                     7-149
PHBDF$               #5-60      5-64
PLBDF$               #5-59      5-67
PUTAD                #5-60    13-468    15-592
RAND               #13-468    13-468   #13-470    13-470   #13-475    13-475   #15-568    15-568   #15-573    15-573
                   #15-592    15-592
REJ$                 #5-77    10-285    10-298
RESMAP               #5-90    13-483    15-593
RETURN               8-177    10-272    10-286    10-299    11-313    11-327    11-339    11-353    12-365    12-376
                    12-392    12-400    12-414    13-484    14-525    15-594    16-631    17-654
SAVMAP               #5-84    13-443    15-567
SLTDF$               #5-60      5-65
SOB                 14-517    15-579
STATE$               #5-59     9-191    #9-194    #9-197    #9-200    #9-203    #9-206    #9-209    #9-212    #9-216
                    #9-219    #9-223    #9-226    #9-229    #9-232    #9-235    #9-238    #9-241    #9-244    #9-248
                    #9-252    #9-255
SVCDF$               #5-60      5-66
TRAN$                #5-59    #9-192    #9-195    #9-198    #9-201    #9-204    #9-207    #9-210    #9-213    #9-214
                    #9-217    #9-220    #9-221    #9-224    #9-227    #9-230    #9-233    #9-236    #9-239    #9-242
                    #9-245    #9-246    #9-249    #9-250    #9-253
XPDDB$               #5-59      5-63
```

```
Symbol table

A$$CHK= 000000          LF.REA= 010000          M$$MGE= 000000          X2HLMX= 001130          Z.MAP   000020
A$$CPS= 000000          LF.SER= 000040          M$$NET= 000000          X2PDF   000000R     002 Z.NAM   000004
A$$PRI= 000000          LF.TIM= 000010          M$$OVR= 000000          X2PKW   000000RG    003 Z.PCB   000012
A$$TRP= 000000          LF.UNL= 020000          N$$ACC= 000001          X2PRM   000036R     002 Z.SCH   000007
CERR  = ****** GX       LF.X2P= 000000          N$$BUF= 000001          X2PST   000000RG    002 $ALPHA= 000022
CFGBF = ****** GX       LN.CLO= 000000          N$$LDV= 000001          X2RTMX= 001130          $ANY  = 000020
CFGSZ = ****** GX       LN.DUM= 000005          N$$MCP= 000001          X2UNT   000003R         $BLANK= 000006
CFLIN = ****** GX       LN.LOA= 000004          N$$MLL= 000001          X2WNMN= 000001          $CAT5 = ****** GX
CHKDEV  000316R         LN.LOO= 000003          N$$MOV= 000010          X2WNMX= 000007          $DIGIT= 000024
C$$CKP= 000000          LN.OAU= 000003          N$$NCT= 000001          X2.CTL  000244R         $DNUMB= 000014
C$$ORE= 000400          LN.OFF= 000001          N$$PEM= 000001          X2.DEV  000216R         $EOS  = 000012
C$$RSH= 177564          LN.ON = 000000          P$$P45= 000000          X2.HLD  000564R         $ERR1T= ****** GX
D$$BUG= 177514          LN.OOP= 000004          P$$WRD= 000000          X2.MBS  000446R         $ERR2C= ****** GX
D$$ISK= 000000          LN.OPE= 000001          Q$$OPT= 000010          X2.MRT  000474R         $ERR2F= ****** GX
D$$L11= 000001          LN.REF= 000002          RTSPC = ****** GX       X2.MWS  000404R         $ERR2I= ****** GX
D$$YNC= 000000          LN.SER= 000002          R$$DER= 000000          X2.RTR  000526R         $ERR2J  000004R
D$$YNM= 000000          LN.STA= 000017          R$$K11= 000001          X2.UNT  000274R         $ERR2K  000046R
E$$XPR= 000000          LN.SUB= 000360          R$$SND= 000000          ZF.COU= 001000          $ERR2L  000106R
FMT8B = ****** GX       LN.TRI= 000006          R$$11M= 000000          ZF.DDM= 000001          $EXIT = 000000
FM.8B = 000000          L$$ASG= 000000          SF.ACT= 000200          ZF.DIA= 004000          $FAIL = 177777
F$$LVL= 000001          L$$DRV= 000000          SF.ENA= 000100          ZF.DLC= 000002          $LAMDA= 000000
G$$TPP= 000000          L$$P11= 000001          SF.LPB= 000004          ZF.DVP= 100000          $NUMBR= 000002
G$$TSS= 000000          L$$11R= 000000          SF.MFL= 000040          ZF.INI= 040000          $PDVTA= ****** GX
G$$TTK= 000000          L.COST  000015          SF.PAC= 000020          ZF.KMX= 000020          $QX2P   000144RG
G$$WRD= 000000          L.CTL   000012          SF.REA= 000010          ZF.LLC= 000004          $RAD50= 000016
I$$RAR= 000000          L.CVA   177776          SF.SER= 000001          ZF.LMC= 000100          $SLTA = ****** GX
I$$RDN= 000000          L.DDM   000002          SF.SVC= 000002          ZF.MAN= 020000          $STRNG= 000004
K$$CNT= 177546          L.DDS   000004          SF.UNL= 000040          ZF.MFL= 000010          $SUBXP= 000010
K$$CSR= 177546          L.DLC   000003          SYNERR= ****** GX       ZF.MTM= 000400          $X2HLD= ****** GX
K$$LDC= 000000          L.DLM   000006          S$$WRG= 000000          ZF.MUX= 000040          $X2MBS= ****** GX
K$$TPS= 000074          L.DLS   000010          S$$YSZ= 007600          ZF.PSE= 002000          $X2MRT= ****** GX
LD$LP = 000000          L.FLG   000000          S.COST  000001          ZF.SLI= 010000          $X2MWS= ****** GX
LF.ACT= 100000          L.KRBA  000016          S.FLG   000000          ZF.TIM= 000200          $X2RET= ****** GX
LF.BRO= 000400          L.LEN = 000022          S.LEN   000004          ZF.X3P= 000000          $$$FLG= 177777
LF.BWT= 000007          L.MPF   000022          S.NMST  000002          ZS.ASN= 100000          $$$KEY= 000000
LF.ENA= 002000          L.NMST  000020          S.OWNR  000003          ZS.BSY= 140000          $$$STA= 000000
LF.LPB= 001000          L.NSTA  000014          T$$KMG= 000000          Z.AVL   000014          $$$TMP= 000000R      004
LF.MDC= 000100          L.OWNR  000021          T$$MIN= 000000          Z.DAT   000016          .PNUMB= ****** GX
LF.MFL= 004000          L.UNT   000013          V$$CTR= 001000          Z.DSP   000000          .PNUMH= ****** GX
LF.MTP= 000020          M$$CRB= 000124          X$$DBT= 000000          Z.FLG   000010          .PSTCN= ****** GX
LF.PAC= 000200          M$$CRX= 000000          X2CTL   000002R         Z.LEN = 000016          .PSTPT= ****** GX
LF.RDY= 040000          M$$FCS= 000000          X2DEV   000000R         Z.LLN   000006          .TPARS= ****** GX


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        000622    001   (RW,I,LCL,REL,CON)
$STATE  000102    002   (RW,D,LCL,REL,CON)
$KTAB   000002    003   (RW,D,LCL,REL,CON)
$KSTR   000007    004   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 15400 Words  ( 61 Pages)
```

```
     85                                       .SBTTL  ERROR MESSAGES
     86
     87                               ;
     88                               ; ERROR MESSAGES
     89                               ;
     90                                       .ENABL  LC
     91 000012                                NTLER$  ,2B,8B,CERR,RTSPC,CFLIN,<default block size>
     92 000046                                NTLER$  ,2D,8,CERR,RTSPC,CFLIN,<Default block size exceeds maximum>
     93 000122                                NTLER$  ,2E,8B,CERR,RTSPC,CFLIN,<default window size>
     94 000156                                NTLER$  ,2G,8,CERR,RTSPC,CFLIN,<Default window size exceeds maximum>
     95 000232                                NTLER$  ,2H,8B,CERR,RTSPC,CFLIN,<timer value>
     96                                       .DSABL  LC
     97
     98 000000                                .PSECT
     70                                          .SBTTL  LOCAL DATA
     71                                 ;
     72                                 ; LOCAL DATA
     73                                 ;
     74 000000                          CURUNM: .BLKW   1                       ; UNMAPPED ADDRESS OF CURRENT BLOCK
```

```
**FILE**ID**VFPCUG




VV      VV   FFFFFFFFFF   PPPPPPPP       CCCCCCCC   UU      UU     GGGGGGGG
VV      VV   FFFFFFFFFF   PPPPPPPP       CCCCCCCC   UU      UU     GGGGGGGG
VV      VV   FF           PP      PP   CC           UU      UU   GG
VV      VV   FF           PP      PP   CC           UU      UU   GG
VV      VV   FF           PP      PP   CC           UU      UU   GG
VV      VV   FF           PP      PP   CC           UU      UU   GG
VV      VV   FFFFFFFF     PPPPPPPP     CC           UU      UU   GG
VV      VV   FFFFFFFF     PPPPPPPP     CC           UU      UU   GG
VV      VV   FF           PP           CC           UU      UU   GG  GGGGGG
VV      VV   FF           PP           CC           UU      UU   GG  GGGGGG
  VV  VV     FF           PP           CC           UU      UU   GG      GG       ....
  VV  VV     FF           PP           CC           UU      UU   GG      GG       ....
    VV       FF           PP             CCCCCCCC   UUUUUUUUUU     GGGGGG         ....
    VV       FF           PP             CCCCCCCC   UUUUUUUUUU     GGGGGG         ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                          .SBTTL  CKCUG1 - Check for unique cug name (old format)                  ;[TD001]
                                                                                                                   ;[TD001]
                                 ;+                                                                                ;[TD001]
                                 ;                                                                                 ;[TD001]
                                 ; *** CKCUG1 - Check for unique cug name (old format)                             ;[TD001]
                                 ;                                                                                 ;[TD001]
                                 ; INPUTS:                                                                         ;[TD001]
                                 ;        CUGNAM - Specified cug name                                              ;[TD001]
                                 ;        CUGNML - Length of CUG name specified                                    ;[TD001]
                                 ;                                                                                 ;[TD001]
                                 ; OUTPUTS:                                                                        ;[TD001]
                                 ;        Carry clear - specified cug name is unique                               ;[TD001]
                                 ;        Carry set - specified cug name is not unique                             ;[TD001]
                                 ;                                                                                 ;[TD001]
                                 ;-                                                                                 ;[TD001]
000504                           CKCUG1:  CALL    $SAVAL                  ; Save all registers                     ;[TD001]
000510  016700  000000G                   MOV     $PSIPT,R0               ; Get address of PSI home block          ;[TD001]
000514  062700  000010                    ADD     #H$CUG,R0               ; Point to CUG listhead                  ;[TD001]
000520                                    GETRV   R0,#2                   ; Get listhead                           ;[TD001]
000536                                    SAVMAP                          ; Save current mapping                   ;[TD001]
000542  016700  000000G          10$:     MOV     $BFR,R0                 ; Get address of next block in list      ;[TD001]
000546  000241                            CLC                             ; Assume success                         ;[TD001]
000550  001432                            BEQ     25$                     ; Branch if end of list (name unique)    ;[TD001]
000552                                    CEACC$  R0                      ; Convert to mapped address              ;[TD001]
000562                                    GETAD   R0,#G$LEN               ; Read in CUG block                      ;[TD001]
000600  026767  000012G 000000G           CMP     G$DTE+$BFR,DTEDES       ; Same DTE as new block?                 ;[TD001]
000606  001355                            BNE     10$                     ; No, so don't check name                ;[TD001]
000610  012702  000006                    MOV     #CUNMMX,R2              ; Get length of cug name                 ;[TD001]
000614  012701  000001'                   MOV     #CUGNAM,R1              ; Point to CUG name                      ;[TD001]
000620  012703  000002G                   MOV     #$BFR+G$NAM,R3          ; Point to CUG name block                ;[TD001]
000624  122123                   20$:     CMPB    (R1)+,(R3)+             ; Is the name unique?                    ;[TD001]
000626  001345                            BNE     10$                     ; Branch if yes                          ;[TD001]
000630  005302                            DEC     R2                      ; More to check?                         ;[TD001]
000632  003374                            BGT     20$                     ; Branch if yes                          ;[TD001]
000634  000261                            SEC                             ; Else error - set C-bit                 ;[TD001]
000636                           25$:     RESMAP                          ; Restore previous mapping               ;[TD001]
000642                           30$:     RETURN                                                                   ;[TD001]
```

```
     64                                                  .SBTTL  LOCAL SYMBOL DEFINITIONS
     65
     66                                         ;
     67                                         ; LOCAL SYMBOLS
     68                                         ;
     69          000040                                  DSCMX = 32.               ; MAXIMUM NUMBER OF DIGITS FOR CALL DATA
     70          000006                                  CUNMMX = 6                ; Maximum length of old-format CUG name          ;[TD001]
```

```
VFPDSA     CREATED BY  MACRO  ON 29-JUN-85 AT 02:34     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME        REFERENCES

ASL$              #5-61      14-334
CALL               8-112     11-203     11-211     11-225     12-232     12-246     13-278     14-293     14-306
DBGTPT            #9-128     #9-137     #9-138     #9-139     #9-143     #9-144     #9-145     #9-149     #9-153     #9-157
                  #9-161     #10-176    #10-186
EMSG$R            #5-61       8-121     11-216     12-240     12-255     13-284     14-299     14-312
ISTAT$            #5-61       9-128
MTRAN$            #9-128
NTLER$            #5-61       7-77       7-78       7-79       7-80       7-81       7-82
RESRG             #5-61      14-340
RETURN             8-116     11-204     11-212     11-227     12-236     12-251     12-269     13-280     14-295     14-308
                   14-341
SAVRG             #5-61      14-314
STATE$            #5-61       9-132     #9-135     #9-141     #9-147     #9-151     #9-155     #9-159     #9-163     10-171
                  #10-174    #10-178    #10-181    #10-184    #10-188
TRAN$             #5-61      #9-133     #9-136     #9-137     #9-138     #9-139     #9-142     #9-143     #9-144     #9-145
                  #9-148     #9-149     #9-152     #9-153     #9-156     #9-157     #9-160     #9-161     #9-164     #10-172
                  #10-175    #10-176    #10-179    #10-182    #10-185    #10-186
VNPDBG            #5-62       8-106
```

```
                                  ;+
                                  ; *** DSNACT - Check account
                                  ;
                                  ; INPUTS
                                  ;          STRBUF - Contains parsed string (minus delimiters etc)
                                  ;          STRLEN - Number of characters in string
                                  ;
                                  ; OUTPUTS
                                  ;          Account checked and stored
                                  ;          Error condition if invalid account
                                  ;-

000222                            DSNACT:
000222  116701  000040'                   MOVB    STRLEN,R1               ; Get length of string
000226  020127  000020                    CMP     R1,#MXACAC              ; Check if account too long
000232  101011                            BHI     101$                    ; Branch if error
000234  110167  000000G                   MOVB    R1,RNAL                 ; Move account length to data area
000240  012700  000041'                   MOV     #STRBUF,R0              ; Move account to data area
000244  012702  000000G                   MOV     #RNACCT,R2
000250                                    CALL    MOVE
000254                                    RETURN

000256                            101$:   EMSG$R  MC                      ; Illegal destination access data
```

```
                                          .SBTTL  LOOK FOR DST$DF MACRO

                                  ;+
                                  ; $QDST - LOOK FOR DST$DF MACRO
                                  ;
                                  ; INPUTS:
                                  ;        NONE
                                  ;
                                  ; OUTPUTS:
                                  ;        C-BIT=SUCCESS/FAILURE
                                  ;        R3,R4,R5=DESTROYED
                                  ;-
                                                                                                       ;[TD001]
 000000                           $QDST::                                                              ;[TD001]
 000000  012705  000000'                  MOV     #DSTDF,R5       ; STATE TABLE ADDRESS                ;[TD001]
 000004  005001                           CLR     R1              ; FULL KEYWORD MATCH LENGTH          ;**-1
 000006  005067  000000G                  CLR     DSFLG           ; INITIALIZE FLAGS WORD
 000012  012702  000000'                  MOV     #DSTKW,R2       ; KEYWORD TABLE ADDRESS
 000016  016703  000000G                  MOV     CFGSZ,R3        ; RECORD LENGTH
 000022  012704  000000G                  MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
 000026  005067  000000G                  CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
 000032                                   CALL    .TPARS          ; GO DO THE PARSE
 000036  103003                           BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
 000040  005367  000000G                  DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
 000044  001401                           BEQ     101$            ; IF EQ, YES
 000046                           20$:    RETURN

                                  ;
                                  ; ERROR CONDITION
                                  ;
 000050                           101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
                                 .SBTTL  FNDLEN - Find total length of variable byte fields            ;[TD001]
                         ;+                                                                            ;[TD001]
                         ; ***   FNDLEN - Find total length of variable byte fields                    ;[TD001]
                         ;                                                                             ;[TD001]
                         ; FUNCTION                                                                    ;[TD001]
                         ;       Given table pointing to counts (one byte) of variable                 ;[TD001]
                         ;       fields, add all lengths to R1.                                        ;[TD001]
                         ;                                                                             ;[TD001]
                         ; INPUTS                                                                      ;[TD001]
                         ;       R0 - Address of table of pointers to counts                           ;[TD001]
                         ;       R1 - Length of fixed fields                                           ;[TD001]
                         ;                                                                             ;[TD001]
                         ; OUTPUTS                                                                     ;[TD001]
                         ;       R1 - Total length of all fields, fixed and variable                   ;[TD001]
                         ;       R0, R2, R3 destroyed                                                  ;[TD001]
                         ;-                                                                            ;[TD001]
001626                   FNDLEN:                                                                       ;[TD001]
001626  012002           5$:     MOV     (R0)+,R2                ; get next table entry                ;[TD001]
001630  001403                   BEQ     20$                     ; zero entry at table end             ;[TD001]
001632  111203                   MOVB    (R2),R3                 ; get length byte                     ;[TD001]
001634  060301                   ADD     R3,R1                   ; add to total count                  ;[TD001]
001636  000773                   BR      5$                      ; loop                                ;[TD001]
                         ;                                                                             ;[TD001]
001640                   20$:    RETURN                                                                ;[TD001]
```

```
	.TITLE	VFPDTE - INITIAL CONFIG FILE SCAN ACTION ROUTINES
	.IDENT	/V05.00/						;[TD001]
	.ENABL	LC							;**-1
;
;
; COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985  BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; NTL - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	02-MAR-82
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;



```

```
                                                .SBTTL  CHKDTE - CHECK FOR VALID DTE ADDRESS

                                        ;+
                                        ;
                                        ; CHKDTE - CHECK FOR VALID DTE ADDRESS
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       CARRY CLEAR - DTE ADDRESS IS UNIQUE
                                        ;           R4 = COUNT OF DTES DEFINED ALREADY
                                        ;       CARRY SET - DTE ADDRESS IS NOT UNIQUE
                                        ;
                                        ;-
001042                                  CHKDTE: CALL    $SAVAL                  ; SAVE ALL REGISTERS
001046  016700  000000G                         MOV     $PSIPT,R0               ; POINT TO PSI HOME BLOCK
001052  062700  000002                          ADD     #H$LDTE,R0              ; POINT TO LOCAL DTE LISTHEAD
001056                                          GETRV   R0,#2                   ; READ IN LISTHEAD
001074                                          SAVMAP                          ; SAVE CURRENT MAPPING
001100  005004                                  CLR     R4                      ; INITIALISE DTE COUNT
001102  016700  000000G                 10$:    MOV     $BFR,R0                 ; POINT TO NEXT BLOCK IN LIST
001106  000241                                  CLC                             ; ASSUME SUCCESS
001110  001431                                  BEQ     30$                     ; BR IF END OF LIST
001112  005204                                  INC     R4                      ; ONE MORE BLOCK SEEN
001114                                          CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
001124                                          GETAD   R0,#L$LEN               ; READ IN DTE DESCRIPTOR
001142  012701  000010                          MOV     #<DTEALN+1>/2,R1        ; GET DTE ADDRESS LENGTH
001146  012702  000000G                         MOV     #$BFR,R2                ; GET ADDRESS OF LOCAL DTE DESCRIPTOR
001152  062702  000020                          ADD     #L$DTEA,R2              ; POINT TO DTE ADDRESS
001156  012703  000017'                         MOV     #DTEPCK,R3              ; POINT TO SPECIFIED DTE ADDRESS
001162  122223                          20$:    CMPB    (R2)+,(R3)+             ; DO THE ADDRESSES MATCH?
001164  001346                                  BNE     10$                     ; BR IF NO: UNIQUE SO FAR
001166  005301                                  DEC     R1                      ; MORE OF ADDRESS TO CHECK?
001170  003374                                  BGT     20$                     ; BR IF YES
001172  000261                                  SEC                             ; INDICATE DUPLICATE ADDRESS
001174                                  30$:    RESMAP                          ; RESTORE PREVIOUS MAPPING
001200  010466  000012                          MOV     R4,R$R4(SP)             ; RETURN COUNT IN TASK R4
001204                                  40$:    RETURN
```

```
     56                                          .SBTTL  MACRO DEFINITIONS
     57
     58                                  ;
     59                                  ; LIBRARY MACROS
     60                                  ;
     61                                          .MCALL  PHBDF$,EMSG$R,SAVRG,RESRG,NTLER$,PUTRC,GETRV
     62                                          .MCALL  ISTAT$,STATE$,TRAN$,PUTAD,GETAD,CEACC$
     63                                          .MCALL  VNPDBG                                                                  ;[TD001]
     64
     65                                  ;
     66                                  ; LIBRARY SYMBOLS
     67                                  ;
     68 000000                                   PHBDF$                  ; DEFINE PSI HOME BLOCK OFFSETS
     69
     70                                  ;
     71                                  ; LOCAL MACRO DEFINITIONS
     72                                  ;
     73                                          .MACRO  SAVMAP
     74                                          MOV     $RBLCK,-(SP)
     75                                          .ENDM
     76
     77                                          .MACRO  RESMAP
     78                                          MOV     (SP)+,$RBLCK
     79                                          .ENDM
     80
```

```
                                                  M 15
VFPPSN - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:37  Page 15-1
Symbol table

ALLPRT  000756R        H$DST   000012         L$$P11= 000001         PSNNML  000006R   002  $ERRD2  000424R   002
ALLPSV  001112R        H$D29   000014         L$$11R= 000000         PSNSHI  000440R        $ERRPA  000310R   002
A$$CHK= 000000         H$FLG   000000         MAXID = 000020         PSNSLO  000352R        $ERRP1  000030R   002
A$$CPS= 000000         H$GLEN  000104         MOVE  = ****** GX      PSNST   000000RG  003  $ERRP2  000066R   002
A$$PRI= 000000         H$GLT   000044         M$$CRB= 000124         PSVTSZ= 000400         $ERRP3  000114R   002
A$$TRP= 000000         H$GNAM  000050         M$$CRX= 000000         P$$P45= 000000         $ERRP4  000144R   002
CERR  = ****** GX      H$GNML= 000020         M$$FCS= 000000         P$$WRD= 000000         $ERRP5  000170R   002
CFGBF = ****** GX      H$GPT   000046         M$$MGE= 000000         Q$$OPT= 000010         $ERRP6  000242R   002
CFGSZ = ****** GX      H$HITS  000034         M$$NET= 000000         RTSPC = ****** GX      $ERR1T= ****** GX
CFLIN = ****** GX      H$HLEN  000044         M$$OVR= 000000         R$$DER= 000000         $EXIT = 000000
CHKSUB  000406R        H$LBDA  000070         N$$ACC= 000001         R$$K11= 000001         $FAIL = 177777
C$$CKP= 000000         H$LBDN  000072         N$$BUF= 000001         R$$SND= 000000         $GETRV= ****** GX
C$$ORE= 000400         H$LDTE  000002         N$$LDV= 000001         R$$11M= 000000         $LAMDA= 000000
C$$RSH= 177564         H$LEN   000042         N$$MCP= 000001         SPACE = 000040         $NUMBR= 000002
D$$BUG= 177514         H$LOTS  000032         N$$MLL= 000001         SPFILL= ****** GX      $PFLAG= ****** GX
D$$ISK= 000000         H$NETW  000024         N$$MOV= 000010         SUBAMX= ****** GX      $PSIPT= ****** GX
D$$L11= 000001         H$NML = 000006         N$$NCT= 000001         SYNERR= ****** GX      $PUTAD= ****** GX
D$$YNC= 000000         H$NPT   000022         N$$PEM= 000001         SYSFDB= ****** GX      $PUTRC= ****** GX
D$$YNM= 000000         H$PTB   000020         PF.PSV= ****** GX      S$$WRG= 000000         $QPSN   000000RG
END     000104R   003  H$PVC   000006         PF.XDF= ****** GX      S$$YSZ= 007600         $RADDR= ****** GX
E$$XPR= 000000         H$RDTE  000004         PF.XGA= ****** GX      T$$KMG= 000000         $RAD50= 000016
FMT10 = ****** GX      H$RNW   000042         PNAMMX= 000006         T$$MIN= 000000         $RBLCK= ****** GX
FMT8  = ****** GX      H$SVC   000036         PORT    000146R        V$$CTR= 001000         $SAVAL= ****** GX
FM.10 = 000000         H$TRB   000016         PORTMX= 000400         X$$DBT= 000000         $STRNG= 000004
FM.8  = 000000         H$XAVL  000100         PORTNO= ****** GX      $ALL16= ****** GX      $SUBXP= 000010
F$$LVL= 000001         H$XBIA  000074         PSHI    000076R   003  $ALPHA= 000022         $$$FLG= 177777
F.RSIZ= ****** GX      H$X29C  000040         PSLO    000054R   003  $ALPOL= ****** GX      $$$KEY= 000000
G$$TPP= 000000         I$$RAR= 000000         PSNDF   000000R   003  $ANY  = 000020         $$$R  = 000010
G$$TSS= 000000         I$$RDN= 000000         PSNEND  000466R        $BFR  = ****** GX      $$$STA= 000000
G$$TTK= 000000         K$$CNT= 177546         PSNFG   000176R        $BLANK= 000006         $$$TMP= 000000R   005
G$$WRD= 000000         K$$CSR= 177546         PSNFLG  000000R   002  $CEACC= ****** GX      .PNUMB= ****** GX
HF$DLM= 000002         K$$LDC= 000000         PSNHI   000004R   002  $DIGIT= 000024         .PNUMH= ****** GX
HF$GWY= 000010         K$$TPS= 000074         PSNKW   000000RG  004  $DNUMB= 000014         .PSTCN= ****** GX
HF$HOS= 000004         LD$LP = 000000         PSNLO   000002R   002  $EOS  = 000012         .PSTPT= ****** GX
HF$XDF= 000020         L$$ASG= 000000         PSNNAM  000007R   002  $ERRD1  000366R   002  .TPARS= ****** GX
H$CUG   000010         L$$DRV= 000000         PSNNM   000052R

. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        001300    001   (RW,I,LCL,REL,CON)
DATA    000462    002   (RW,D,LCL,REL,CON)
$STATE  000114    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13395 Words  ( 53 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:40.03
SY:VFPPSN.V2,[132,134]VFPPSN/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPPSN
```

```
                                        .SBTTL  TPARS STATE TABLES
                                ;
                                ; TPARS STATE TABLES
                                ;
 000056                                 ISTAT$  PVCST,PCVKW

                                ;
                                ; PVC$DF STATE TABLES
                                ;
 000056                                 STATE$  PVCDF
 000056                                  TRAN$   %PVC$DF%,,,1,SYNERR

 000056                                 STATE$
 000056                                  TRAN$   $RAD50,,PVNAM           ; CHANNEL IDENTIFICATION

 000056                                 STATE$
 000056                                  TRAN$   !CHKDLM,$EXIT           ; IGNORE IF DLM CIRCUIT
 000056                                  TRAN$   <',>

 000056                                 STATE$  CHNLNO
 000056                                  TRAN$   $NUMBR,,PVLCN           ; LOGICAL CHANNEL NUMBER

 000056                                 STATE$
 000056                                  TRAN$   <',>

 000056                                 STATE$
 000056                                  TRAN$   $NUMBR,,PVCT            ; COUNTER TIMER

 000056                                 STATE$
 000056                                  TRAN$   <',>

 000056                                 STATE$
 000056                                  TRAN$   "OFF",PSZ       ; STATE
 000056                                  TRAN$   "ON",PSZ

 000056                                 STATE$  PSZ
 000056                                  TRAN$   !END,$EXIT,PVEND
 000056                                  TRAN$   <',>

 000056                                 STATE$
 000056                                  TRAN$   $NUMBR,WNDSZ,PVPSZ      ; PACKET SIZE
 000056                                  TRAN$   $LAMDA

 000056                                 STATE$  WNDSZ
 000056                                  TRAN$   !END,$EXIT,PVEND
 000056                                  TRAN$   <',>

 000056                                 STATE$
 000056                                  TRAN$   $NUMBR,OWNER,PVWSZ      ; WINDOW SIZE
 000056                                  TRAN$   $LAMDA

 000056                                 STATE$  OWNER
 000056                                  TRAN$   !END,$EXIT,PVEND
 000056                                  TRAN$   <',>

 000056                                 STATE$
 000056                                  TRAN$   $RAD50,PFLAGS,PVOWN     ; PROCESS OWNER
```

```
                                                                          N  1
**FILE**ID**VFGX2P






VV        VV    FFFFFFFFFF        GGGGGGGG      XX        XX      222222        PPPPPPPP
VV        VV    FFFFFFFFFF        GGGGGGGG      XX        XX      222222        PPPPPPPP
VV        VV    FF              GG              XX        XX    22        22    PP        PP
VV        VV    FF              GG              XX        XX    22        22    PP        PP
VV        VV    FF              GG                 XX   XX                 22    PP        PP
VV        VV    FF              GG                 XX   XX                 22    PP        PP
VV        VV    FFFFFFFF        GG                    XX                  22      PPPPPPPP
VV        VV    FFFFFFFF        GG                    XX                  22      PPPPPPPP
VV        VV    FF              GG   GGGGGG        XX   XX           22          PP
VV        VV    FF              GG   GGGGGG        XX   XX           22          PP
   VV   VV      FF              GG        GG      XX        XX      22            PP                   ....
   VV   VV      FF              GG        GG      XX        XX      22            PP                   ....
     VV         FF                GGGGGG        XX        XX    2222222222      PP                   ....
     VV         FF                GGGGGG        XX        XX    2222222222      PP                   ....




LL                SSSSSSSS      TTTTTTTTTT
LL                SSSSSSSS      TTTTTTTTTT
LL              SS                   TT
LL              SS                   TT
LL              SS                   TT
LL              SS                   TT
LL                SSSSSS             TT
LL                SSSSSS             TT
LL                        SS         TT
LL                        SS         TT
LL                        SS         TT
LL                        SS         TT
LLLLLLLLLL      SSSSSSSS             TT
LLLLLLLLLL      SSSSSSSS             TT
```

```
VFGX2P - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 02:31  Page 11-3
Symbol table

Size of core pool: 16552 Words  ( 63 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:33.42
SY:VFGX2P.V2,[132,134]VFGX2P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFGX2P
```

```
                                                .SBTTL  $QX3P - LOOK FOR X3P$DF MACRO

                                        ;+
                                        ;
                                        ; $QX3P - LOOK FOR X3P$DF MACRO
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       ALL REGISTERS DESTROYED
                                        ;
                                        ;-
 000000  012705  000000'                $QX3P:: MOV     #X3PDF,R5       ; STATE TABLE ADDRESS
 000004  005001                                 CLR     R1              ; FULL KEYWORD MATCH LENGTH
 000006  012702  000000'                        MOV     #X3PKW,R2       ; KEYWORD TABLE ADDRESS
 000012  016703  000000G                        MOV     CFGSZ,R3        ; RECORD LENGTH
 000016  012704  000000G                        MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
 000022  005067  000000G                        CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
 000026                                         CALL    .TPARS          ; GO DO THE PARSE
 000032  103003                                 BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
 000034  005367  000000G                        DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
 000040  001401                                 BEQ     101$            ; IF EQ, YES
 000042                                 20$:    RETURN
                                        ;
                                        ; ERRORS
                                        ;
 000044                                 101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
VFIND - VNP FIND ENTITY ROUTINE MACRO V05.03b Saturday 29-Jun-85 02:33  Page 7
$FNOBJ - FIND OBJECT

     76                                                   .SBTTL  $FNOBJ - FIND OBJECT
     77                                          ;+
     78                                          ; *** - $FNOBJ - FIND OBJECT
     79                                          ;
     80                                          ; INPUTS:
     81                                          ;        R1 = OBJECT TYPE
     82                                          ;
     83                                          ; OUTPUTS:
     84                                          ;        C-BIT = SUCCESS/FAILURE
     85                                          ;        R0 = UNMAPPED ADDRESS OF OBJECT BLOCK, IF FOUND
     86                                          ;        R1 = UNMAPPED ADDRESS OF PREVIOUS BLOCK
     87                                          ;-
     88
     89 000002                                   $FNOBJ::
     90 000002  012700  000000G                          MOV     #$OBJHD,R0              ; GET START OF OBJECTS LIST
     91 000006  011067  000000G                          MOV     (R0),$BFR               ; GET FIRST OBJECT IN LIST
     92 000012  010146                                   MOV     R1,-(SP)                ; SAVE OBJECT NUMBER ON STACK
     93 000014  010067  177760                           MOV     R0,CURUNM               ; SAVE UNMAPPED ADDR OF CURRENT BLOCK
     94 000020  016701  177754                   10$:    MOV     CURUNM,R1               ; SAVE ADDRESS OF PREVIOUS
     95 000024  016767  000000G 177746                   MOV     $BFR,CURUNM             ; GET NEXT OBJECT IN LIST
     96 000032  000261                                   SEC                             ; ASSUME FAILURE
     97 000034  001426                                   BEQ     40$                     ; IF EQ, NO MORE
     98 000036                                           CEACC$  CURUNM,R0               ; CONVERT TO MAPPED ADDRESS
     99 000050                                   20$:    GETAD   R0,#O.LEN               ; READ OBJECT IN INTERNAL BUFFER
    100 000066  032767  000004  000000G                  BIT     #QF$KNO,$QUAL           ; IS THIS REQUEST FOR KNOWN OBJECTS ?
    101 000074  001003                                   BNE     30$                     ; IF NE, YES
    102 000076  121667  000002G                          CMPB    (SP),$BFR+O.TYP         ; IS THIS THE RIGHT OBJECT ?
    103 000102  001346                                   BNE     10$                     ; IF NE, NO
    104 000104  016700  177670                   30$:    MOV     CURUNM,R0               ; GET UNMAPPED ADDRESS OF BLOCK
    105 000110  000241                                   CLC                             ; ELSE, INDICATE SUCCESS
    106 000112  005226                           40$:    INC     (SP)+                   ; CLEAN UP STACK
    107 000114                                           RETURN
```

Table of contents

```
     5-   56    MACRO DEFINITIONS
     6-   80    LOCAL SYMBOL DEFINITIONS
     7-   89    LOCAL DATA
     8-  109    ERROR MESSAGES
     9-  120    LOOK FOR CUG$DF MACRO
    10-  151    CUG$DF STATE TABLES
    11-  188    CUG$DF ACTION ROUTINES
    14-  279    CKCUG0 - Check for unique cug name (new format)              ;[TD001]
    15-  319    CKCUG1 - Check for unique cug name (old format)              ;[TD001]
    16-  361    ALCUG0 - Allocate a CUG name block (new format)              ;[TD001]
    17-  380    ALCUG1 - Allocate a CUG name block (old format)              ;[TD001]
    18-  398    ALCUG - Allocate and fill a CUG name block (general)         ;[TD001]
```

```
                                        .IF EQ  G$NML                                                   ;[TD001]
                                        .ERROR  ; Code requires G$NML to have non-zero offset           ;[TD001]
                                        .ENDC                                                           ;[TD001]
                                                                                                        ;[TD001]
                                        .SBTTL  ALCUGO - Allocate a CUG name block (new format)         ;[TD001]
                                ;+                                                                      ;[TD001]
                                ; *** ALCUGO - Allocate a CUG name block (new format)                   ;[TD001]
                                ;                                                                       ;[TD001]
                                ; INPUTS                                                                ;[TD001]
                                ;       All parsed fields of CUG$DF macro                               ;[TD001]
                                ;                                                                       ;[TD001]
                                ; OUTPUTS                                                               ;[TD001]
                                ;       Carry clear - CUG name block allocated and linked into CUG chain;[TD001]
                                ;       Carry set - Insufficient store to allocate CUG name block       ;[TD001]
                                ;-                                                                      ;[TD001]
000644                          ALCUGO:                                                                 ;[TD001]
000644  012700  000015                  MOV     #G$NML,R0               ; Offset to contain CUG name length ;[TD001]
000650  012701  000016                  MOV     #G$GNAM,R1              ; Offset at which to store CUG name ;[TD001]
000654  116702  000000'                 MOVB    CUGNML,R2               ; Length of CUG name ...        ;[TD001]
000660  062702  000016                  ADD     #G$LEN,R2               ; + fixed length = size of block ;[TD001]
000664                                  CALL    ALCUG                   ; allocate and fill block       ;[TD001]
000670                                  RETURN                                                          ;[TD001]
```

```
     72                                           .SBTTL  ERROR MESSAGES
     73
     74 000000                                    .PSECT  DATA,D
     75                                           .NLIST  BEX
     76                                                                                                        ;**-1
     77 000000                                    NTLER$  ,N2,10,CERR,RTSPC,CFLIN,<CUG name>
     78 000022                                    NTLER$  ,N8,10,CERR,RTSPC,CFLIN,<remote dte address>
     79 000056                                    NTLER$  ,N9,10,CERR,RTSPC,CFLIN,<low subaddress>
     80 000106                                    NTLER$  ,M0,10,CERR,RTSPC,CFLIN,<high subaddress>
     81 000136                                    NTLER$  ,M2,10,CERR,RTSPC,CFLIN,<call data value>
     82 000166                                    NTLER$  ,M3,10,CERR,RTSPC,CFLIN,<call data mask>
     83
     84                                           .EVEN                                                        ;**-1
     85                                           .LIST   BEX
     86 000000                                    .PSECT
```

N 8

```
**FILE**ID**VFPDSN




VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD       SSSSSSSS   NN      NN
VV      VV   FFFFFFFFFF   PPPPPPPP     DDDDDDDD       SSSSSSSS   NN      NN
VV      VV   FF           PP      PP   DD      DD   SS           NN      NN
VV      VV   FF           PP      PP   DD      DD   SS           NN      NN
VV      VV   FF           PP      PP   DD      DD   SS           NNNN    NN
VV      VV   FF           PP      PP   DD      DD   SS           NNNN    NN
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD     SSSSSS     NN  NN  NN
VV      VV   FFFFFFFF     PPPPPPPP     DD      DD     SSSSSS     NN  NN  NN
VV      VV   FF           PP           DD      DD           SS   NN    NNNN
VV      VV   FF           PP           DD      DD           SS   NN    NNNN
  VV  VV     FF           PP           DD      DD           SS   NN      NN        ....
  VV  VV     FF           PP           DD      DD           SS   NN      NN        ....
    VV       FF           PP           DDDDDDDD     SSSSSSSS     NN      NN        ....
    VV       FF           PP           DDDDDDDD     SSSSSSSS     NN      NN        ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                          .SBTTL  Parser Subexpression Action Routines
                                  ;+
                                  ; *** CLRSTR - Clear string length field
                                  ;
                                  ; INPUTS
                                  ;       None
                                  ;
                                  ; OUTPUTS
                                  ;       STRLEN cleared
                                  ;-
000264  105067  000040'           CLRSTR: CLRB    STRLEN
000270                                    RETURN
```

```
                                        .SBTTL  TPARS STATE TABLES
                                ;
                                ; TPARS STATE TABLES
                                ;
000056                                  ISTAT$  DSTST,DSTKW

                                ;
                                ; DST$DF STATE TABLES
                                ;
000056                                  STATE$  DSTDF
000056                                   TRAN$   %DST$DF%,,,1,SYNERR

000056                                  STATE$
000056                                   TRAN$   ''X25'',DSNM,,DS.X25,DSFLG        ; TYPE
000056                                   TRAN$   ''X29'',,,DS.X29,DSFLG

000056                                  STATE$  DSNM
000056                                   TRAN$   <',>

000056                                  STATE$
000056                                   TRAN$   $RAD50,,DSNAM                     ; DESTINATION NAME          ;[TD001]
                                                                                                               ;**-1
000056                                  STATE$
000056                                   TRAN$  <',>

000056                                  STATE$
000056                                   TRAN$   $NUMBR,,DSPRI                     ; PRIORITY

000056                                  STATE$
000056                                   TRAN$   <',>

000056                                  STATE$
000056                                   TRAN$   $NUMBR,,DSOBJ                     ; OBJECT NUMBER

000056                                  STATE$
000056                                   TRAN$   !END,$EXIT
000056                                   TRAN$   <',>

000056                                  STATE$
000056                                   TRAN$   $RAD50,,DSTSK                     ; TASK NAME

000056                                  STATE$  END
000056                                   TRAN$   $EOS,$EXIT
000056                                   TRAN$   <';>,$EXIT

000056                                  STATE$
```

```
                                .SBTTL  BCDLEN - Find total length of variable BCD fields             ;[TD001]
                        ;+                                                                            ;[TD001]
                        ; ***   BCDLEN - Find total length of variable BCD fields                     ;[TD001]
                        ;                                                                             ;[TD001]
                        ; FUNCTION                                                                    ;[TD001]
                        ;       Given table pointing to digit counts (one byte) of variable           ;[TD001]
                        ;       BCD fields, add byte lengths to R1.                                   ;[TD001]
                        ;                                                                             ;[TD001]
                        ; INPUTS                                                                      ;[TD001]
                        ;       R0 - Address of table of pointers to counts                           ;[TD001]
                        ;       R1 - Length of fixed fields                                           ;[TD001]
                        ;                                                                             ;[TD001]
                        ; OUTPUTS                                                                     ;[TD001]
                        ;       R1 - Total length of all fields, fixed and variable                   ;[TD001]
                        ;       R0, R2, R3 destroyed                                                  ;[TD001]
                        ;-                                                                            ;[TD001]
001642                  BCDLEN:                                                                       ;[TD001]
001642  012002          5$:     MOV     (R0)+,R2                ; get next table entry                ;[TD001]
001644  001405                  BEQ     20$                     ; zero entry at table end             ;[TD001]
001646  111203                  MOVB    (R2),R3                 ; get length byte                     ;[TD001]
001650  006203                  ASR     R3                      ; convert from digit count to byte count;[TD001]
001652  005503                  ADC     R3                                                            ;[TD001]
001654  060301                  ADD     R3,R1                   ; add to total count                  ;[TD001]
001656  000771                  BR      5$                      ; loop                                ;[TD001]
                        ;                                                                             ;[TD001]
001660                  20$:    RETURN                                                                ;[TD001]
```

```
     56                                        .SBTTL  MACRO DEFINITIONS
     57
     58                                ;
     59                                ; LIBRARY MACROS
     60                                ;
     61                                        .MCALL  EMSG$R,SLTDF$,SAVRG,RESRG,DTEDF$,GETAD,PUTAD,CEACC$
     62                                        .MCALL  PHBDF$,NTLER$,ISTAT$,TRAN$,STATE$,GETRV
     63                                        .MCALL  VNPDBG                                                          ;[TD001]
     64                                ;
     65                                ; LIBRARY SYMBOLS
     66                                ;
     67 000000                                 DTEDF$                          ; LOCAL DTE DESCRIPTOR BLOCK OFFSETS
     68 000000                                 PHBDF$                          ; DEFINE PSI HOME BLOCK OFFSETS
     69 000000                                 SLTDF$                          ; DEFINE SLT OFFSETS
     70                                ;
     71                                ; LOCAL MACRO DEFINITIONS
     72                                ;
     73                                        .MACRO  SAVMAP
     74                                        MOV     $RBLCK,-(SP)
     75                                        .ENDM
     76
     77                                        .MACRO  RESMAP
     78                                        MOV     (SP)+,$RBLCK
     79                                        .ENDM
```

```
                                          .SBTTL  CHKDEV - CHECK VALIDITY OF LINE-ID

                                  ;+
                                  ; **-CHKDEV-CHECK VALIDITY OF LINE-ID
                                  ;
                                  ; THIS ROUTINE MAKES SURE THAT THE DEVICE SPECIFIED IS KNOWN TO THE SYSTEM,
                                  ; AND THAT THE FIELDS SPECIFIED APPLY.
                                  ;
                                  ; INPUTS:
                                  ;       LINNAM = DDM NAME OF LINE IN RAD50
                                  ;       LINCTL = CONTROLLER NUMBER
                                  ;       LINUNT = UNIT NUMBER
                                  ;
                                  ; OUTPUTS:
                                  ;       IF CC, LINE IS FOUND AND IS LEGAL
                                  ;           R4 = SYSTEM LINE NUMBER
                                  ;       ELSE, UNRECOGNIZED LINE
                                  ;
                                  ;-

001206                            CHKDEV:
001206                                    CALL    $SAVAL                  ; SAVE ALL REGISTERS

                                  ; MAKE SURE THE PROCESS EXISTS, AND IS A DDM
                                  ;
001212  016700  000000G                   MOV     LINNAM,R0               ; GET DDM NAME TO LOOK FOR
001216                                    CALL    FNDPDV                  ; FIND IT'S PDV ADDRESS
001222  103433                            BCS     80$                     ; IF CC, LINE NOT IN SYSTEM
                                  ;
                                  ; MAKE SURE THE LINE EXISTS
                                  ;
001224  012704  177777                    MOV     #-1,R4                  ; CALCULATE SYSTEM LINE NUMBER
001230  016700  000000G                   MOV     $SLTNM,R0               ; GET NUMBER OF SYSTEM LINES
001234  016703  000000G                   MOV     $SLTMA,R3               ; GET ADDRESS OF SYSTEM LINE INDEX TABLE
001240  012302                    10$:    MOV     (R3)+,R2                ; GET ADDR OF SYSTEM LINE TABLE ENTRY
001242  005204                            INC     R4                      ; UPDATE SYSTEM LINE NUMBER
001244  126201  000002                    CMPB    L.DDM(R2),R1            ; DOES THIS PROCESS MATCH ?
001250  001010                            BNE     20$                     ; IF NE, NO - KEEP LOOKING
001252  126267  000012  000000G           CMPB    L.CTL(R2),LINCTL        ; DOES THIS CONTROLLER NUMBER MATCH ?
001260  001004                            BNE     20$                     ; IF NE, NO - KEEP LOOKING
001262  126267  000013  000000G           CMPB    L.UNT(R2),LINUNT        ; DOES THIS UNIT NUMBER MATCH ?
001270  001403                            BEQ     30$                     ; IF EQ, FOUND THE CORRECT SLN
001272  005300                    20$:    DEC     R0                      ; LOOK THROUGH ALL SYSTEM LINES
001274  001361                            BNE     10$                     ; ...
001276  000405                            BR      80$                     ; IF HERE, LINE NOT FOUND
001300  032762  000020  000000    30$:    BIT     #LF.MTP,L.FLG(R2)       ; IS THIS A MULTI-DROP LINE?
001306  001001                            BNE     80$                     ; IF NE, YES - ERROR
001310  000401                            BR      100$                    ; ELSE VALID LINE ID
001312  000261                    80$:    SEC                             ; SET CARRY
001314  010466  000012            100$:   MOV     R4,R$R4(SP)             ; SET USER R4 TO SLN
001320                            110$:   RETURN                          ; AND RETURN TO USER MODE
```

```
     82                                                  .SBTTL  LOCAL SYMBOLS
     83
     84          000006                                  PNAMMX = 6              ; Max characters in network name [old format]   ;[TD001]
     85          000020                                  MAXID = 16.             ; Max length of net. mgmt. 'id-string'          ;[TD001]
     86          000400                                  PORTMX = 256.           ; MAXIMUM NUMBER OF PORTS                       ;[TD001]
     87          000400                                  PSVTSZ = 256.           ; Number of entries in each PSV mapping table   ;[TD001]
     88          000040                                  SPACE  = 40             ; ASCII SPACE                                   ;[TD001]
```

```
                                                                      N 15
VFPPSN      CREATED BY  MACRO  ON 29-JUN-85 AT 02:37       PAGE 1
SYMBOL  CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE               REFERENCES

ALLPRT     000756 R           13-348     #14-383
ALLPSV     001112 R           13-355     #15-420
CERR     = ******   GX        8-105       8-106       8-107       8-108       8-109       8-110       8-111       8-112       8-113
CFGBF    = ******   GX        9-137
CFGSZ    = ******   GX        9-136
CFLIN    = ******   GX        8-105       8-106       8-107       8-108       8-109       8-110       8-111       8-112       8-113
CHKSUB     000406 R           12-273     #12-287      13-302
FMT10    = ******   GX        8-105       8-106       8-107       8-108       8-112       8-113
FMT8     = ******   GX        8-109       8-110       8-111
FM.10    = 000000            #8-105      #8-106      #8-107      #8-108      #8-112      #8-113
FM.8     = 000000            #8-109      #8-110      #8-111
F.RSIZ   = ******   GX       *13-343     *14-398     *14-403     *15-432     *15-437     *15-439
HF$DLM   = 000002             12-277      13-307      13-339
HF$GWY   = 000010             11-246
HF$HOS   = 000004             11-243      13-315
HF$XDF   = 000020             11-249      11-251
H$FLG      000000            *13-338
H$GLEN     000104             13-319      15-439
H$GLT      000044            *15-440
H$GNAM     000050             13-333
H$GPT      000046            *15-441
H$HITS     000034            *13-341
H$HLEN     000044             13-317
H$LEN      000042             13-312
H$LOTS     000032            *13-340
H$NETW     000024             13-329
H$NPT      000022            *13-337
H$PTB      000020             14-401
MAXID    = 000020            #6-85        7-100      11-211      11-213      13-332
MOVE     = ******   GX        11-219
N$$VCT   = ******             14-390      15-431      15-436
PF.PSV   = ******   GX        11-248      13-353
PF.XDF   = ******   GX        11-253      11-254      13-313      13-330
PF.XGA   = ******   GX        11-245
PNAMMX   = 000006            #6-84        11-256      13-328
PORT       000146 R          #11-228
PORTMX   = 000400            #6-86        11-232
PORTNO   = ******   GX       *11-234      13-337      14-384      14-392      14-396
PSNDF      000000 R            9-133
PSNEND     000466 R          #13-311
PSNFG      000176 R          #11-239
PSNFLG     000000 R          #7-96       *11-220     *11-241      11-243      11-246      11-249      11-251     *12-277     *13-307
                              13-315      13-338      13-339
PSNHI      000004 R          #7-98       *11-222     *12-276     *13-306      13-341
PSNKW      000000 RG           9-135     #10-154
PSNLO      000002 R          #7-97       *11-221     *12-275      13-304      13-340
PSNNAM     000007 R          #7-100       13-327
PSNNM      000052 R          #11-206
PSNNML     000006 R          #7-99        11-209      11-216      11-256
PSNSHI     000440 R          #13-302
PSNSLO     000352 R          #12-273
PSNST      000000 RG         #10-154
```

```
B  1
C  1
D  1
E  1
F  1
G  1
H  1
I  1
J  1
K  1
L  1
M  1
N  1   VFGX2P
B  2
C  2
D  2
E  2
F  2
G  2
H  2
I  2
J  2
K  2
L  2
M  2
N  2
B  3
C  3
D  3
E  3
F  3
G  3   VFGX3P
H  3
I  3
J  3
K  3
L  3
M  3
N  3
B  4
C  4
D  4
E  4
F  4
G  4
H  4
I  4   VFIND
J  4
K  4
L  4
M  4
N  4
B  5
C  5
D  5
E  5
F  5
G  5
H  5
I  5
```

```
J  5
K  5
L  5
M  5   VFPCUG
N  5
B  6
C  6
D  6
E  6
F  6
G  6
H  6
I  6
J  6
K  6
L  6
M  6
N  6
B  7
C  7
D  7
E  7
F  7
G  7
H  7
I  7   VFPDSA
J  7
K  7
L  7
M  7
N  7
B  8
C  8
D  8
E  8
F  8
G  8
H  8
I  8
J  8
K  8
L  8
M  8
N  8   VFPDSN
B  9
C  9
D  9
E  9
F  9
G  9
H  9
I  9
J  9
K  9
L  9
M  9
N  9
B 10
C 10
D 10
```

```
E 10
F 10
G 10   VFPDST
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12   VFPDTE
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14   VFPPSN
K 14
L 14
```

```
M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15
B 16
C 16
D 16
E 16   VFPPVC
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16
```

```
    243 000056                                    TRAN$   $LAMDA
    244
    245 000056                                   STATE$  PFLAGS
    246 000056                                    TRAN$   !END,$EXIT,PVEND
    247 000056                                    TRAN$   <',>
    248
    249 000056                                   STATE$
    250 000056                                    TRAN$   $NUMBR,END,PVFG         ; FLAGS BYTE
    251 000056                                    TRAN$   $LAMDA
    252
    253 000056                                   STATE$  END
    254 000056                                    TRAN$   $EOS,$EXIT
    255 000056                                    TRAN$   <';>,$EXIT
    256                                   ;
    257                                   ; CHECK FOR DLM CIRCUIT NAME - SUBEXPRESSION
    258                                   ;
    259 000056                                   STATE$  CHKDLM
    260 000056                                    TRAN$   '-,,PVDLM
    261
    262 000056                                   STATE$
    263 000056                                    TRAN$   $DNUMB,,PVCTL
    264
    265 000056                                   STATE$
    266 000056                                    TRAN$   '.
    267
    268 000056                                   STATE$
    269 000056                                    TRAN$   $DNUMB,,PVSTA,PF.DLM,FLAG
    270
    271 000056                                   STATE$
    272 000056                                    TRAN$   <',>,$EXIT
    273
    274 000056                                   STATE$
```

```
VFPPVC -
PVDLM -
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
```

C 1

```
                                        .SBTTL  PVDLM - CHECK FOR DLM PVC

                                ;+
                                ;
                                ; PVDLM - CHECK FOR DLM PVC
                                ;
                                ; INPUTS:
                                ;       PVCNAM - PROCESS NAME (ASCII)
                                ;
                                ; OUTPUTS:
                                ;       DLMPDV - DLM'S PDV ADDRESS
                                ;
                                ;-
002276  012700  000027'         PVDLM:  MOV     #PVCNAM,R0      ; POINT TO SPECIFIED PROCESS NAME
002302                                  CALL    $CAT5           ; CONVERT TO RAD50
002306  020127  015355                  CMP     R1,#^RDLM       ; IS THIS DLM?
002312  001011                          BNE     INVCIR          ; BR IF NO - ILLEGAL CIRCUIT ID
002314  010100                          MOV     R1,R0           ; GET DLM'S RAD50 PROCESS NAME

002316                                  CALL    FNDPDV          ; GET DLM'S PDV INDEX
002322  103405                          BCS     INVCIR          ; BR IF NOT FOUND
002324  110167  000052'         10$:    MOVB    R1,DLMIND       ; SAVE PDV INDEX
002330  010067  000016'                 MOV     R0,DLMPDV       ; SAVE DLM'S PDV ADDRESS
002334                                  RETURN
                                ;
                                ; ERRORS
                                ;
002336                          INVCIR: EMSG$R  NO              ; ILLEGAL CIRCUIT ID
                                ;
                                ; PVCTL - GET CONTROLLER NUMBER FOR DLM
                                ;
                                        .ENABL  LSB
002344  116767  000000G 000047' PVCTL:  MOVB    .PNUMB,DLMCTL   ; SAVE CONTROLLER NUMBER
002352  000403                          BR      10$             ; CHECK FOR VALIDITY
002354  116767  000000G 000051' PVSTA:  MOVB    .PNUMB,DLMSTA   ; SAVE STATION NUMBER
002362  005767  000000G         10$:    TST     .PNUMH          ; LEGAL VALUE?
002366  001363                          BNE     INVCIR          ; BR IF NO - ERROR
002370  105767  000001G                 TSTB    .PNUMB+1        ; LEGAL VALUE?
002374  001360                          BNE     INVCIR          ; BR IF NO
002376                                  RETURN
                                        .DSABL  LSB

        000001                          .END
```

```
                    .SBTTL  ERROR MESSAGES

                    .ENABL  LC

000036              NTLER$  ,L0,10,CERR,RTSPC,CFLIN,<remote dte name>
000066              NTLER$  ,L1,10,CERR,RTSPC,CFLIN,<remote dte address>
000122              NTLER$  ,L2,8,CERR,RTSPC,CFLIN,<Remote dte block allocation failure>

                    .DSABL  LC
                    .LIST   BEX
                    .EVEN
000000              .PSECT
```

```
    50                                          .SBTTL  Local Symbols
    51
    52          000042                          DQUOTE = <'">                   ; String delimiter
    53          000054                          COMMA = <',>                    ; Operand separator
    54          000073                          COMENT = <';>                   ; Start-of-comment character
    55          000074                          BRA = 74                        ; Left angle bracket (<)
    56          000076                          KET = 76                        ; Right angle bracket (>)
    57
    58          000006                          MXNOID = 6                      ; Max length of node id
    59          000020                          MXACUS = 16.                    ; Max length of access cntrl user
    60          000010                          MXACPA = 8.                     ; Max length of access cntrl password
    61          000020                          MXACAC = 16.                    ; Max length of access cntrl account
    62          000020                          MAXID = 16.                     ; Max length of network name
    63
    64          000020                          MAXSTR = 16.                    ; Max length of string seen in parse
```

```
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
```

```
                                   ;+
                                   ; *** RNAACT - Check account
                                   ;
                                   ; INPUTS
                                   ;       STRBUF - Contains parsed string (minus delimiters etc)
                                   ;       STRLEN - Number of characters in string
                                   ;
                                   ; OUTPUTS
                                   ;       Account checked and stored
                                   ;       Error condition if invalid account
                                   ;-

000312                             RNAACT:
000312  116701  000164'                    MOVB    STRLEN,R1               ; Get length of string
000316  020127  000020                     CMP     R1,#MXACAC              ; Check if account too long
000322  101011                             BHI     101$                    ; Branch if error
000324  110167  000000G                    MOVB    R1,RNAL                 ; Move account length to data area
000330  012700  000165'                    MOV     #STRBUF,R0              ; Move account to data area
000334  012702  000000G                    MOV     #RNACCT,R2
000340                                     CALL    MOVE
000344                                     RETURN

000346                             101$:   EMSG$R  RC                      ; Illegal access data
```

```
**FILE**ID**VFPX29
```

```
VV      VV   FFFFFFFFFF   PPPPPPPP     XX      XX     222222       999999
VV      VV   FFFFFFFFFF   PPPPPPPP     XX      XX     222222       999999
VV      VV   FF           PP      PP   XX      XX   22      22   99      99
VV      VV   FF           PP      PP   XX      XX   22      22   99      99
VV      VV   FF           PP      PP     XX  XX             22   99      99
VV      VV   FF           PP      PP     XX  XX             22   99      99
VV      VV   FFFFFFFF     PPPPPPPP         XX             22     99999999
VV      VV   FFFFFFFF     PPPPPPPP         XX             22     99999999
VV      VV   FF           PP             XX  XX       22                 99
VV      VV   FF           PP             XX  XX       22                 99
  VV  VV     FF           PP           XX      XX   22                   99       ....
  VV  VV     FF           PP           XX      XX   22                   99       ....
    VV       FF           PP           XX      XX   2222222222     999999         ....
    VV       FF           PP           XX      XX   2222222222     999999         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
**FILE**ID**VFPX3P




VV      VV   FFFFFFFFFF   PPPPPPPP     XX      XX     333333     PPPPPPPP
VV      VV   FFFFFFFFFF   PPPPPPPP     XX      XX     333333     PPPPPPPP
VV      VV   FF           PP      PP   XX      XX   33      33   PP      PP
VV      VV   FF           PP      PP   XX      XX   33      33   PP      PP
VV      VV   FF           PP      PP     XX  XX             33   PP      PP
VV      VV   FF           PP      PP     XX  XX             33   PP      PP
VV      VV   FFFFFFFF     PPPPPPPP         XX             33     PPPPPPPP
VV      VV   FFFFFFFF     PPPPPPPP         XX             33     PPPPPPPP
VV      VV   FF           PP             XX  XX             33   PP
VV      VV   FF           PP             XX  XX             33   PP
  VV  VV     FF           PP           XX      XX   33      33   PP                ....
  VV  VV     FF           PP           XX      XX   33      33   PP                ....
    VV       FF           PP           XX      XX     333333     PP                ....
    VV       FF           PP           XX      XX     333333     PP                ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

VFPX3
Symbo

SY:VF

B 8

```
VFPX3P - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:41  Page 12-3
Symbol table

SY:VFPX3P.V2,[132,134]VFPX3P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPX3P
```

```
VHELP -

     55
     56
     57
     58
     59
     60
     61
     62
     63
     64
     65
     66
     67
     68
     69
```

C 8

```
     55                                        ;****
     56                                        ; MACRO CALLS
     57                                        ;****
     58
     59                                                .MCALL  VNPDF$,CALLR
     60
     61                                        ;****
     62                                        ; LOCAL DATA
     63                                        ;****
     64
     65                                        ;
     66                                        ; VNP OFFSET DEFINITIONS
     67                                        ;
     68 000000                                         VNPDF$
     69
```

```
000276  001402                            BEQ     41$              ; IF NOT - BRANCH
000300  066716  000000G                   ADD     $RSIZE,(SP)      ; ADD SIZE OF SECOND READ
000304                           41$:     SWTCHI  (SP)+            ; IS NEW DESTINATION ADDRESS
000310  066767  000000G 000000G           ADD     $RSIZE,$RADDR    ; ADJUST SOURCE ADDRESS
000316  166705  000000G                   SUB     $RSIZE,R5        ; DETERMINE REMAINING CETAB SIZE

000322  022705  001000           14$:     CMP     #512.,R5         ; ONE TRANSFER LEFT ?
000326  002006                            BGE     13$              ; IF YES - BRANCH
000330  012767  001000  000000G           MOV     #512.,$RSIZE     ; SET UP TRANSFER SIZE
000336  005267  000000G                   INC     $BIGCT           ; SET FLAG FOR ANOTHER TRANSFER
000342  000404                            BR      16$              ; CONTINUE

000344  010567  000000G          13$:     MOV     R5,$RSIZE        ; SET FOR READ OF CETAB REMAINING
000350  005067  000000G                   CLR     $BIGCT           ; INIT CETAB > 2000 BYTES FLAG

000354                           16$:     GETRV                    ; AND READ IT IN
000360  005767  000000G                   TST     $BIGCT           ; ANY MORE LEFT ?
000364  001401                            BEQ     12$              ; IF NOT - BRANCH
000366  000735                            BR      40$              ; CONTINUE



                                 ;
                                 ; RELOCATE THE SLT ADDRESSES
                                 ;
000370                           12$:
                                          .IF DF  R$$MPL                                                ;[MP01]

                                          MOV     $CETST,$RADDR    ; GET ACTUAL START OF CETAB          ;[MP01]
                                          ADD     $CETLN,$RADDR    ; CALCULATE ADDRESS OF NEXT TRANSFER ;[MP01]
                                          MOV     #$CETAB-2,R5     ; GET START OF CETAB-2               ;[MP01]
                                          ADD     $CETLN,R5        ; GET END OF CETAB                   ;[MP01]
                                          CLR     R2               ; INIT REG                           ;[MP01]
                                          MOVB    (R5),R2          ; GET NUMBER OF CONTROLLERS FOR LAST ;[MP01]
                                                                   ; KRB TABLE EXPANSION                ;[MP01]
                                          ADD     #2,R5            ; ADJUST ADDRESS OF TRANSFER         ;[MP01]
                                          ASLB    R2               ; CONVERT TO BYTES                   ;[MP01]
                                          MOV     R2,$RSIZE        ; SET TRANSFER SIZE                  ;[MP01]
                                          ADD     R2,$CETLN        ; UPDATE TOTAL SIZE FOR VIMO         ;[MP01]
                                          SWTCHI  R5               ; SET UP ADDRESS OF TRANSFER         ;[MP01]
                                          GETRV                    ; READ LAST KRB TABLE INTO VNP TEMPLATE ;[MP01]

                                          .ENDC                                                         ;[MP01]

000370  005067  000000G                   CLR     $BIGCT           ; INIT LARGE CETAB FLAG
000374  016701  000000G                   MOV     $SLTMA,R1        ; GET START ADDRESS OF SLT MAPPING TABLE
000400  016702  000000G                   MOV     $SLTNM,R2        ; GET NUMBER OF SLT ENTRIES
000404  001002                            BNE     15$              ; Branch if non-zero                         ;[TD001]
000406  000167  000434                    JMP     120$             ; IF EQ, SOMETHING SMELLS                    ;[TD001]
000412  160021                   15$:     SUB     R0,(R1)+         ; SUBTRACT SHIFT BIAS
000414  005302                            DEC     R2
000416  003375                            BGT     15$              ; LOOP TILL DONE

                                 ; READ IN IMAGE OF FIRST BLOCK OF NETWORK POOL ( HOME BLOCK )

000420                                    SWTCHI  #$NTLHB          ; POINT AT BUFFER

                                 ; SET UP POINTERS FOR 32.-WORD BLOCK MANIPULATIONS
```

```
    000500                              RETURN


            000001                      .END
```

```
                                        ;+
                                        ;       **-VSYMB
                                        ;
                                        ;       OPEN EXEC AND COMEXEC SYMBOL TABLES AND DEFINE
                                        ;       SYMBOLS NECESSARY TO JOB AT HAND.
                                        ;
                                        ; INPUTS:
                                        ;       FILE    - CONTAINS NAME OF SYSTEM IMAGE FILE ( ASCIZ )
                                        ;                         THIS NAME CAN BE SPECIFIED AS SIMPLY SY:
                                        ;                         AND THE USERS' CURRENT UIC WILL BE USED
                                        ;                         AS A DEFAULT. RSX11M IS THE FILENAME
                                        ;                         DEFAULT.
                                        ;
                                        ; OUTPUTS:
                                        ;       ALL DEFINED SYMBOLS ARE MOVED TO THE ROOT SECTION WHERE
                                        ;       THEY CAN BE REFERENCED BY ANY OVERLAY.
                                        ;
                                        ;-

001340  012700  000000G         VSYMB:  MOV     #FILE,R0        ; ADDRESS OF USER RESPONSE
001344  010002                          MOV     R0,R2           ; POINTER TO END OF DEV:[XX,YY]
001346  005302                          DEC     R2              ; ...
001350  010001                          MOV     R0,R1           ; COPY ADDRESS

001352  105711                  10$:    TSTB    (R1)            ; EOS ?
001354  001415                          BEQ     40$             ; YES
001356  121127  000056                  CMPB    (R1),#'.        ; END OF FILENAME ?
001362  001422                          BEQ     50$             ; YES, USER SPECIFIED FILENAME
001364  121127  000135                  CMPB    (R1),#']        ; LAST CHARACTER UIC ?
001370  001001                          BNE     20$             ; NO ...
001372  010102                          MOV     R1,R2           ; POINT AT CHARACTER
001374  121127  000072          20$:    CMPB    (R1),#':        ; LAST CHARACTER DEVICE NAME
001400  001001                          BNE     30$             ; NO ...
001402  010102                          MOV     R1,R2           ; POINT AT CHARACTER
001404  005201                  30$:    INC     R1              ; INCRMENT POINTER
001406  000761                          BR      10$             ; LOOP

001410  005202                  40$:    INC     R2              ; GET TRUE DEV/UIC STRING POINTER
001412  020200                          CMP     R2,R0           ; WAS ANY DEV/UIC ENTERED ?
001414  001406                          BEQ     60$             ; NO, THEN ASSUME FILENAME
001416  020201                          CMP     R2,R1           ; WAS FILE-NAME TYPED ?
001420  001004                          BNE     60$             ; YES
001422  012701  000000G                 MOV     #$RSX,R1        ; USE DEFAULT FILE NAME
001426  000411                          BR      80$

001430  005202                  50$:    INC     R2              ; FORM TRUE EOS POINTER
001432  105011                  60$:    CLRB    (R1)            ; TERMINATE STRING HERE
001434  016704  000000G                 MOV     $CLBUF,R4       ; DESTINATION FOR FILENAME
001440  010201                          MOV     R2,R1           ; COPY END OF DEV/UIC POINTER
001442  112124                  70$:    MOVB    (R1)+,(R4)+     ; MOVE NAME
001444  001376                          BNE     70$             ; ...

                                        .IF DF  R$$MPL                                          ;[MP01]

                                        MOV     R2,R1           ; POINT AT NAME
                                        MOV     #$RSX,R4        ; POINT AT BUFFER
                                75$:    MOVB    (R1)+,(R4)+     ; COPY FILENAME
                                        BNE     75$             ; IF MORE - BRANCH
                                        ;+
                                        ;       -VCHECK         - CHECK GEN PARTITION BOUNDARY CONDITIONS
                                        ;
                                        ;       ISSSUE WARNING OR FATAL ERROR MESSAGE DEPENDING ON THE
                                        ;       RELATIONSHIP BETWEEN THE GEN PARTITION AND THE HIGHEST
                                        ;       BLOCK NUMBER ALLOCATED IN THE SYSTEM IMAGE FILE RSX11M.SYS.
                                        ;
                                        ;-

 004130 016767  000000G 000000G VCHECK: MOV     $PARHD,$BFR             ; GET PARTITION LISTHEAD
 004136 016767  000000G 000000G         MOV     .PLGTH,$RSIZE           ; LOAD PCB LENGTH FOR VIRTUAL I/O
 004144 016767  000000G 000000G 10$:    MOV     $BFR,$RADDR             ; GET ADDRESS OF NEXT PCB
 004152 001450                          BEQ     101$                    ; NO GEN PARTITION !!
 004154                                 GETRV                           ; READ IN PCB
 004160 026727  000004G 026226          CMP     P.NAM+$BFR,#^RGEN       ; MATCH BOTH
 004166 001366                          BNE     10$                     ;  HALVES OF
 004170 026727  000006G 000000          CMP     P.NAM+2+$BFR,#^R        ;   GEN PARTITION
 004176 001362                          BNE     10$                     ;    NAME ?
                                        ;
                                        ;       CONVERT GEN END ADDRESS TO BLOCK NUMBER ( 64. BYTES )
                                        ;               ALL COMPARISONS ARE MADE IN 32. WORD BLOCKS
                                        ;
 004200 016767  000014G 000000G         MOV     P.REL+$BFR,$GENST       ; GEN START BIAS
 004206 016700  000000G                 MOV     $GENST,R0               ; ...
 004212 016767  000016G 000000G         MOV     P.BLKS+$BFR,$GENND      ; GEN SIZE IN 32. WORD BLOCKS
 004220 066767  000000G 000000G         ADD     $GENST,$GENND           ; GEN ENDING BIAS
 004226 016704  000000G                 MOV     $GENND,R4               ; GEN END BIAS
 004232 016702  000000G                 MOV     $SYSIZ,R2               ; SYSTEM SIZE IN 32. WORD BLOCKS
 004236 010403                          MOV     R4,R3                   ; COPY TO R3
 004240 000241                          CLC                             ; CLEAR C-BIT
 004242 006003                          ROR     R3                      ;  AND DIVIDE BY
 004244 006203                          ASR     R3                      ;   8 TO GET NUMBER
 004246 006203                          ASR     R3                      ;    OF DISK BLOCKS.
 004250 062703  000002                  ADD     #2,R3                   ; TWO BLOCKS FOR LABEL GROUP
 004254 010367  000200G                 MOV     R3,$BFR+200             ; STORE FOR POSSIBLE WARNING

 004260 020200                          CMP     R2,R0                   ; IS ANY PART OF GEN IN SYS IMAGE ?
 004262 101412                          BLOS    102$                    ; NO

 004264 020204                          CMP     R2,R4                   ; DOES GEN SPAN HIBLK ?
 004266 103416                          BLO     103$                    ; YES

 004270 000241                  35$:    CLC                             ; SUCCESS
 004272                         40$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
 004274 012746  004270'         101$:   MOV     #35$,-(SP)              ; FORCE RETURN ADDRESS
 004300                                 ERRPT$  ERR10,$EROUT            ; ISSUE ERROR MESSAGE
 004310 012746  004340'         102$:   MOV     #1033$,-(SP)            ; SET RETURN ADDRESS
 004314                                 ERRPT$  ERR11,$EROUT            ; NO GEN PARTITION

 004324 012746  004340'         103$:   MOV     #1033$,-(SP)            ; SET RETURN ADDRESS
 004330                                 ERRPT$  ERR12,$EROUT

 004340                         1033$:  ERRPT$  ERR14,$EROUT            ; ISSUE WARNING
```

```
                                        ;***
                                        ; LIBRARY MACROS
                                        ;***
                                                .MCALL  GETAD,GETRV,PUTAD,PUTRC,SWTCHI,SWTCHO
                                                .ENABL  LC

                                        ;***
                                        ; LOCAL DATA
                                        ;***

 000000                                         .PSECT  DATA,D

                                        ;
                                        ; LIBRARY BLOCK BIAS
                                        ;
 000000  177777                         .LBIAS::.WORD   -1

                                        ;
                                        ; 32. WORD BUFFER FOR LIBRARY BLOCK
                                        ;
 000002                                 .LBUFF::.BLKW   32.

 000000                                         .PSECT
```

```
    251 000272                                       ERROR$  GETFAL,WARN      ; Else, failed to get command line
    252 000312  005767  000014'             20$:     TST     EBUF             ; Did any error occur?
    253 000316  001231                               BNE     1$               ; If ne, yes
    254 000320                              EOJ:     RETURN
```

```
004052  016767  000000G 000054G          MOV     .PNUMB,RQB+LO.SEG ; STORE SEGMENT SIZE

004060                                   RETURN

004062                          101$:    ERROR$  ILLSEG



                                ;
                                ; Alias node address
                                ;
004102                          $STADD::
004102  005767  000100'                  TST     AREAN           ; Use area default number ?
004106  001024                           BNE     10$             ; If no - branch

004110                                   SWTCHI  #$HMBUF         ; Set up input buffer
004116                                   GETRV   $DECPT,#D$END   ; Read decnet home block
004136                                   SWTCHI                  ; Reset default buffer

004144  016700  000014G                  MOV     $HMBUF+D$LNUM,R0 ; Get executor address
004150  042700  000000C                  BIC     #^CARAMSK,R0    ; Get executor area
004154  010067  000100'                  MOV     R0,AREAN        ; Store default area number

004160  005767  000102'         10$:     TST     ADDRS           ; Check lower address limit
004164  001413                           BEQ     101$            ; If illegal - branch
004166  022767  001777  000102'          CMP     #MXADDR,ADDRS   ; Check high address limit
004174  002407                           BLT     101$            ; If illegal - branch

004176  016767  000102' 000010G          MOV     ADDRS,RQB+LR.DES ; Store node address
004204  056767  000100' 000010G          BIS     AREAN,RQB+LR.DES ; Store area number

004212                                   RETURN
004214                          101$:    ERROR$  ILLNAD          ; Illegal node address

                                ;
                                ; Alias node name
                                ;
004234                          $STALI::
004234  012702  000002G                  MOV     #RQB+LR.ALI,R2  ; Get address of output buffer
004240  012701  000006                   MOV     #MXALIN,R1      ;  and length.
004244  000523                           BR      FORWR           ; Continue in common code

                                ;
                                ; Destination node name
                                ;
004246                          $STDES::
004246  012702  000010G                  MOV     #RQB+LR.DES,R2  ; Get address of output buffer
004252  012701  000006                   MOV     #MXDSTN,R1      ; Size of buffer.
004256  000516                           BR      FORWR           ; And continue in common code

                                ;
                                ; Executor node name
                                ;
004260  032767  000000G 000000G $STEND::BIT      #VF.ADD,$VFLAG  ; Node address specified?
004266  001003                           BNE     10$             ; Br if yes
004270  012702  000012G                  MOV     #LR.NAM+RQB,R2  ; Get address of node name storage
004274  000477                           BR      STRNAM
004276  016767  000072' 000012G 10$:     MOV     NODID,LR.ADD+RQB ; Store node address
```

```
                                              .SBTTL  PVC$DF ACTION ROUTINES

                                      ;
                                      ; PVC CIRCUIT ID
                                      ;
000056                                PVNAM:                                                                       ;[TD001]
000056  012700  000026'                       MOV     #PVCNML,R0      ; Address PVC name buffer                    ;[TD001]
000062  012701  000006                        MOV     #PVNAMX,R1      ; Max length of old-format PVC name          ;[TD001]
000066                                        CALL    STORID          ; Check and store PVC name                   ;[TD001]
000072  103413                                BCS     101$            ; Branch if error                            ;[TD001]
000074  016767  000000G 000010'               MOV     BLKSZD,PVCPSZ   ; SET UP DEFAULT MAXIMUM BLOCK SIZE          ;**-13
000102  016767  000000G 000012'               MOV     WNDSZD,PVCWSZ   ; SET UP DEFAULT WINDOW SIZE
000110  005067  000014'                       CLR     PVCOWN          ; INITIALIZE PROCESS OWNER
000114  105067  000025'                       CLRB    PVCFLG          ; INITIALIZE FLAGS BYTE
000120                                        RETURN
                                      ;
                                      ; ERRORS
                                      ;
000122                                101$:   EMSG$R  NO              ; ILLEGAL CIRCUIT ID

                                      ;
                                      ; LOGICAL CHANNEL NUMBER
                                      ;
000130  005767  000000G               PVLCN:  TST     .PNUMH          ; LEGAL VALUE?
000134  001011                                BNE     101$            ; BR IF NO
000136  016700  000000G                       MOV     .PNUMB,R0       ; GET SPECIFIED LOGICAL CHANNEL NUMER
000142  003406                                BLE     101$            ; BR IF ILLEGAL VALUE
000144  020027  000000G                       CMP     R0,#CHNLMX      ; LEGAL VALUE?
000150  101003                                BHI     101$            ; BR IF NO
000152  010067  000004'                       MOV     R0,PVCLCN       ; STORE LOGICAL CHANNEL NUMBER
000156                                        RETURN
                                      ;
                                      ; ERRORS
                                      ;
000160                                101$:   EMSG$R  09              ; ILLEGAL LOGICAL CHANNEL NUMBER

                                      ;
                                      ; COUNTER TIMER
                                      ;
000166  005767  000000G               PVCT:   TST     .PNUMH          ; LEGAL VALUE?
000172  001004                                BNE     101$            ; BR IF NO
000174  016767  000000G 000006'               MOV     .PNUMB,PVCCTM   ; SAVE COUNTER TIMER VALUE
000202                                        RETURN
                                      ;
                                      ; ERRORS
                                      ;
000204                                101$:   EMSG$R  05              ; ILLEGAL COUNTER TIMER VALUE
```

```
ALLXCB
ALPVC0
ALPVC1
A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
BIASX
BLKSZD=
CERR  =
CFGBF =
CFGSZ =
CFLIN =
CF$BLK=
CHKBLK=
CHKDLM
CHKWND=
CHNLMX=
CHNLNO
CKPVC0
CKPVC1
C$DTE
C$FLG
C$GNAM
C$LCN
C$LEN
C$LNK
C$NAM
C$NML
C$PORT
C$$CKF-
C$$ORE=
C$$RSH=
DLMCTL
DLMIND
DLMPDV
DLMSLT
DLMSTA
DLMUNT
DTEDES=
D$$BUG=
D$$ISK=
D$$L11=
D$$YNC=
D$$YNM=
END
E$$XPR=
FLAG
FMT10 =
FMT8  =
FM.10 =
FM.8  =
FNDHSH
FNDPDV=
FNDPRT
F$$LVL=
F.RSIZ=
```

C 2

```
VFPPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:38  Page 23-1
Symbol table

ALLXCB  000440R           G$$TPP= 000000            M$$FCS= 000000            SYNERR= ****** GX         X$ST    000005
ALPVC0  001152R           G$$TSS= 000000            M$$MGE= 000000            SYSFDB= ****** GX         X$TCLZ  000030
ALPVC1  001344R           G$$TTK= 000000            M$$NET= 000000            S$$WRG= 000000            X$TIMC  000001
A$$CHK= 000000            G$$WRD= 000000            M$$OVR= 000000            S$$YSZ= 007600            X$TIMR  000000
A$$CPS= 000000            HF$DLM= 000002            N$$ACC= 000001            T$$KMG= 000000            X$TXI   000102
A$$PRI= 000000            HF$GWY= 000010            N$$BUF= 000001            T$$MIN= 000000            X$TXQ   000066
A$$TRP= 000000            HF$HOS= 000004            N$$LDV= 000001            V$$CTR= 001000            X$TYP   000015
BIASX   002256R           HF$XDF= 000020            N$$MCP= 000001            WNDSZ   000070R       003 X$USR   000022
BLKSZD= ****** GX         H$HADD= ****** GX         N$$MLL= 000001            WNDSZD= ****** GX         X$WAQ   000062
CERR  = ****** GX         H$HSZ = ****** GX         N$$MOV= 000010            XA$NCO= 000020            X$WSZ   000020
CFGBF = ****** GX         H$CUG   000010            N$$NCT= 000001            XA$RR = 000200            X$$DBT= 000000
CFGSZ = ****** GX         H$DST   000012            N$$PEM= 000001            XA$UCL= 000040            $ALPHA= 000022
CFLIN = ****** GX         H$D29   000014            OWNER   000112R       003 XA$URE= 000100            $ALPOL= ****** GX
CF$BLK= 000102            H$FLG   000000            PCVKW   000000RG      004 XF$NCO= 000020            $ANY  = 000020
CHKBLK= ****** GX         H$GLEN  000104            PFLAGS  000134R       003 XF$OFF= 000200            $BFR  = ****** GX
CHKDLM  000166R       003 H$GLT   000044            PF.DLM= 000001            XF$ORI= 000002            $BLANK= 000006
CHKWND= ****** GX         H$GNAM  000050            PF.PSV= ****** GX         XF$OTI= 000001            $CAT5 = ****** GX
CHNLMX= ****** GX         H$GNML= 000020            PF.XDF= ****** GX         XF$TIL= 000010            $CEACC= ****** GX
CHNLNO  000022R       003 H$GPT   000046            PORTNO= ****** GX         XF$TIR= 000004            $DIGIT= 000024
CKPVC0  000664R           H$HITS  000034            PRTADD  000000R       002 XT$CHN= 000100            $DNUMB= 000014
CKPVC1  001022R           H$HLEN  000044            PRTBIA  000002R       002 XT$DGR= 000002            $EOS  = 000012
C$DTE   000012            H$LBDA  000070            PSZ     000046R       003 XT$FAS= 000200            $ERRN0  000056R       002
C$FLG   000015            H$LBDN  000072            PVCCTM  000006R       002 XT$INC= 000004            $ERRN1  000116R       002
C$GNAM  000016            H$LDTE  000002            PVCDF   000000R       003 XT$OUC= 000010            $ERRN2  000152R       002
C$LCN   000010            H$LEN   000042            PVCFLG  000025R       002 XT$PVC= 000001            $ERRN3  000204R       002
C$LEN   000016            H$LOTS  000032            PVCLCN  000004R       002 X$ABQ   000076            $ERR00= ****** GX
C$LNK   000000            H$NETW  000024            PVCNAM  000027R       002 X$ALF   000055            $ERR05= ****** GX
C$NAM   000002            H$NML = 000006            PVCNML  000026R       002 X$AUC   000056            $ERR09= ****** GX
C$NML   000002            H$NPT   000022            PVCOWN  000014R       002 X$CLEN= 000025            $ERRP8= ****** GX
C$PORT  000014            H$PTB   000020            PVCPRT  000024R       002 X$CTIM  000060            $ERR1T= ****** GX
C$$CKF= 000000            H$PVC   000006            PVCPSZ  000010R       002 X$DIAG  000003            $EXIT = 000000
C$$ORE= 000400            H$RDTE  000004            PVCST   000000RG      003 X$DTE   000024            $FAIL = 177777
C$$RSH= 177564            H$RNW   000042            PVCT    000166R           X$FLG   000014            $GETAD= ****** GX
DLMCTL  000047R       002 H$SVC   000036            PVCTL   002344R           X$GLEN  000106            $GETRV= ****** GX
DLMIND  000052R       002 H$TRB   000016            PVCWSZ  000012R       002 X$LCN   000026            $LAMDA= 000000
DLMPDV  000016R       002 H$XAVL  000100            PVDLM   002276R           X$LEN   000106            $NUMBR= 000002
DLMSLT  000020R       002 H$XBIA  000074            PVEND   000352R           X$MOWN  000023            $PFLAG= ****** GX
DLMSTA  000051R       002 H$X29C  000040            PVFG    000314R           X$NLRE  000052            $PSIPT= ****** GX
DLMUNT  000050R       002 INIXCB  001676R           PVLCN   000130R           X$NNRE  000054            $PUTAD= ****** GX
DTEDES= ****** GX         INVCIR  002336R           PVNAM   000056R           X$NPL   000006            $QPVC   000000RG
D$$BUG= 177514            I$$RAR= 000000            PVNAMX= 000006            X$NPR   000007            $RADDR= ****** GX
D$$ISK= 000000            I$$RDN= 000000            PVOWN   000256R           X$NRBY  000032            $RAD50= 000016
D$$L11= 000001            K$$CNT= 177546            PVPSZ   000212R           X$NRPK  000042            $RBLCK= ****** GX
D$$YNC= 000000            K$$CSR= 177546            PVSTA   002354R           X$NRRE  000053            $SAVAL= ****** GX
D$$YNM= 000000            K$$LDC= 000000            PVWSZ   000234R           X$NTBY  000036            $STRNG= 000004
END     000156R       003 K$$TPS= 000074            P$$P45= 000000            X$NTPK  000046            $SUBXP= 000010
E$$XPR= 000000            LD$LP = 000000            P$$WRD= 000000            X$PKSZ  000016            $$$FLG= 177777
FLAG    000022R       002 LNKEND= ****** GX         Q$$OPT= 000010            X$PR    000011            $$$KEY= 000002
FMT10 = ****** GX         L$$ASG= 000000            RTSPC = ****** GX         X$PRT   000021            $$$R  = 000010
FMT8  = ****** GX         L$$DRV= 000000            R$$DER= 000000            X$PS    000010            $$$STA= 000000
FM.10 = 000000            L$$P11= 000001            R$$K11= 000001            X$RPR   000013            $$$TMP= 000013R       005
FM.8  = 000000            L$$11R= 000000            R$$SND= 000000            X$RPS   000012            .PNUMB= ****** GX
FNDHSH  002120R           MAXID = 000020            R$$11M= 000000            X$RTRY  000002            .PNUMH= ****** GX
FNDPDV= ****** GX         MOVE  = ****** GX         SPACE = 000040            X$RXI   000104            .PSTCN= ****** GX
FNDPRT  001522R           M$$CRB= 000124            SP$P4   000054RG      002 X$RXQ   000072            .PSTPT= ****** GX
F$$LVL= 000001            M$$CPX= 000000            STORID= ****** GX         X$SS    000004            .TPARS= ****** GX
F.RSIZ= ****** GX
```

```
VFPRDT -
LOOK FOR

    113
    114
    115
    116
    117
    118
    119
    120
    121
    122
    123
    124
    125
    126
    127
    128
    129
    130
    131
    132
    133
    134
    135
    136
    137
    138
    139
    140
```

D 2

```
                                     .SBTTL  LOOK FOR RDT$DF MACRO

                              ;+
                              ; $QRDT - LOOK FOR RDT$DF MACRO
                              ;
                              ; INPUTS:
                              ;       NONE
                              ;
                              ; OUTPUTS:
                              ;       C-BIT=SUCCESS/FAILURE
                              ;       R3,R4,R5=DESTROYED
                              ;-
000000  012705  000000'       $QRDT:: MOV     #RDTDF,R5        ; STATE TABLE ADDRESS
000004  005001                        CLR     R1               ; FULL KEYWORD MATCH LENGTH
000006  012702  000000'               MOV     #RDTKW,R2        ; KEYWORD TABLE ADDRESS
000012  016703  000000G               MOV     CFGSZ,R3         ; RECORD LENGTH
000016  012704  000000G               MOV     #CFGBF,R4        ; RECORD BUFFER ADDRESS
000022  005067  000000G               CLR     SYNERR           ; CLEAR SYNTAX ERROR FLAG
000026                                CALL    .TPARS           ; GO DO THE PARSE
000032  103003                        BCC     20$              ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000034  005367  000000G               DEC     SYNERR           ; DID SYNTAX ERROR OCCUR?
000040  001401                        BEQ     101$             ; IF EQ, YES
000042                        20$:    RETURN

                              ;
                              ; ERROR CONDITION
                              ;
000044                        101$:   EMSG$R  1T               ; SYNTAX ERROR
```

```
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40  Page 6
Macro references and definitions

     66                                                      .SBTTL  Macro references and definitions
     67
     68                                                      .MCALL  EMSG$R,NTLER$,STATE$,TRAN$,ISTAT$,RNWDF$,PHBDF$
     69                                                      .MCALL  GETRV,CEACC$,GETAD,PUTAD,SAVMAP,RESMAP
     70                                                      .MCALL  VNPDBG
     71
     72 000000                                               RNWDF$                          ; Remote network block symbol definitions
     73 000000                                               PHBDF$                          ; PSI homeblock symbol definitions
     74                                               ;
     75                                               ; Local macro definitions
     76                                               ;
     77                                                      .MACRO  SAVMAP
     78                                                      MOV     $RBLCK,-(SP)
     79                                                      .ENDM
     80
     81                                                      .MACRO  RESMAP
     82                                                      MOV     (SP)+,$RBLCK
     83                                                      .ENDM
```

```
VFPRNW -
End of m
```

```
                                         .SBTTL  End of macros - check and allocate network block
                                 ;+
                                 ; *** RNWEND - End of macros - check and allocate network block
                                 ;
                                 ; INPUTS
                                 ;       All parsed fields for RNW$DF/RNA$DF
                                 ;
                                 ; OUTPUTS
                                 ;       If network name unique, block allocated and linked
                                 ;       into remote network chain
                                 ;
                                 ;-
000354                           RNWEND::
000354                                   CALL    CKNET                   ; Check network name unique
000360  103404                           BCS     101$                    ; Error if not unique
000362                                   CALL    ALNET                   ; Allocate new network block
000366  103404                           BCS     102$                    ; Error if no space
000370                                   RETURN

000372                           101$:   EMSG$R  RA                      ; Bad network name
000400                           102$:   EMSG$R  RX                      ; Allocation failure
```

C 6

Table of contents

```
     5-    52     MACRO DEFINITIONS
     6-    64     LOCAL DATA STORAGE AND ERROR MESSAGES
     7-    90     LOOK FOR X29$DF MACRO
     8-   120     TPARS STATE TABLES
     9-   151     X29$DF ACTION ROUTINES
```

Table o

```
  5-
  6-
  7-
  8-
  9-
 10-
 11-
 12-
```

D 6

C 7

```
VFPX3P - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:41
Table of contents

     5-   55     MACRO DEFINITIONS
     6-   78     LOCAL SYMBOL DEFINITIONS
     7-   85     LOCAL DATA
     8-  108     LOOK FOR ***$DF MACRO
     9-  144     TPARS STATE TABLES
    10-  177     CHN$DF STATE TABLE
    11-  204     X3P$DF ACTION ROUTINES
    12-  260     CHN$DF ACTION ROUTINES
```

```
VFPX3

SYMBO

SYMBO

BLKDE
BLKMA
BLKSZ
BLKSZ
CERR
CFGBF
CFGSZ
CFLIN
CHEND
CHHI
CHKBL
CHKWN
CHLO
CHNDF
CHNLH
CHNLL
CHNLM
CHNLM
DTEDE
FMT8
FM.8
F.RSI
K$HIC
K$LEN
K$LOC
L$CHL
L$LEN
L$MCH
N$$VC
RTSPC
SYNEF
SYSFD
S$$BA
WNDDE
WNDMA
WNDSZ
WNDSZ
X3PDF
X3PKV
X3PST
$ALL
$ALPH
$ANY
$BFR

$BLAN
$CEAC
$DIG
$DNUM
$EOS
$ERR
$ERR
```

D 7

```
                                                                         C  8
VFPX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:41      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE            REFERENCES

BLKDEF     000060 R        #11-209
BLKMAX     000102 R        #11-221
BLKSZD   = ******   GX     *11-211      11-223
BLKSZM   = ******   GX     *11-225
CERR     = ******   GX      7-100
CFGBF    = ******   GX      8-129
CFGSZ    = ******   GX      8-128
CFLIN    = ******   GX      7-100
CHEND      000302 R        #12-286
CHHI       000254 R        #12-279
CHKBLK   = ******   GX      11-209      11-221
CHKWND   = ******   GX      11-236      11-248
CHLO       000220 R        #12-266
CHNDF      000036 R         8-124
CHNLHI     000002 R        #7-93       *12-273     *12-285      12-293      12-311
CHNLLO     000000 R        #7-92       *12-272      12-292      12-312
CHNLMN   = 000001          #6-83
CHNLMX   = ******   GX      12-270      12-283
DTEDES   = ******   GX      12-296
FMT8     = ******   GX      7-100
FM.8     = 000000          #7-100
F.RSIZ   = ******   GX     *12-294     *12-298     *12-303     *12-305     *12-316
K$HIC      000004          *12-293
K$LEN      000006           12-286      12-294      12-305
K$LOC      000002          *12-292
L$CHLS     000034           12-299      12-302
L$LEN      000122           12-298      12-316
L$MCHN     000036          *12-317
N$$VCT   = ******           12-296
RTSPC    = ******   GX      7-100
SYNERR   = ******   GX     *8-130      *8-133
SYSFDB   = ******   GX     *12-294     *12-298     *12-303     *12-305     *12-316
S$$BAS   = ******           7-100       7-100
WNDDEF     000140 R        #11-236
WNDMAX     000162 R        #11-248
WNDSZD   = ******   GX     *11-238      11-250
WNDSZM   = ******   GX     *11-252
X3PDF      000000 R         8-122
X3PKW      000000 RG        8-127      #9-148
X3PST      000000 RG       #9-148
$ALL16   = ******   GX      12-289
$ALPHA   = 000022          #9-148
$ANY     = 000020          #9-148
$BFR     = ******   GX     *12-291     *12-292     *12-293      12-299     *12-299     *12-301      12-305      12-306     *12-308
                           *12-317
$BLANK   = 000006          #9-148
$CEACC   = ******   GX      12-296
$DIGIT   = 000024          #9-148
$DNUMB   = 000014          #9-148
$EOS     = 000012          #9-148
$ERR00   = ******   GX      11-257
$ERR01   = ******   GX      11-231
```

```
VHELP -
$VHELP
     71
     72
     73
     74
     75
     76
     77
     78
     79
     80
     81
     82
     83
     84
     85
     86
     87
     88
```

D 8

```
                                   .SBTTL  $VHELP - PROCESS HELP COMMAND
                                ;+
                                ; *** - $VHELP - PROCESS HELP COMMAND
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       HELP TEXT PRINTED
                                ;
                                ;-

000000                          $VHELP::
000000  016704  000002G                 MOV     RQB+LR.HLP,R4           ; POINT AT LINE AFTER ''HELP''
000004                                  CALLR   $HELP                   ; PRINT HELP INFO


        000001                          .END
```

```

000426  005767  000000G                 TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM ?
000432  100413                          BMI     17$             ; NO
000434  012767  000000G 000000G         MOV     #$RADDR,$ADRZ   ; THIS HALF SHOULD BE ZEROED
000442  012767  000000G 000000G         MOV     #$RBLCK,$ADRX   ;  AND THIS HALF USED
000450  012767  000001  000000G         MOV     #1,$UNBSZ       ; SIZE OF BLOCKS WHEN CALLING $ALN18
000456  000412                          BR      18$             ; CONTINUE
000460  000554                  35$:    BR      110$            ; CONTINUE

000462  012767  000000G 000000G 17$:    MOV     #$RADDR,$ADRX   ; THIS HALF SHOULD BE USED
000470  012767  000000G 000000G         MOV     #$RBLCK,$ADRZ   ;  AND THIS HALF ZEROED
000476  012767  000100  000000G         MOV     #64.,$UNBSZ     ; SIZE OF BLOCKS WHEN CALLING $ALN18

000504  005077  000000G         18$:    CLR     @$ADRZ          ; ZERO HALF
000510  016777  000000G 000000G         MOV     $NTLPT,@$ADRX   ; AND USE HALF
000516  001405                          BEQ     19$             ; DON'T READ IF NOT SET UP

000520                                  GETAD   ,#64.           ; READ IN IMAGE OF HOME BLOCK


                                ; BIAS MULTIPOINT REVERSE MAPPING TABLE POINTERS ( IF ANY )

000532  016702  000000G         19$:    MOV     $LLCTA,R2       ; GET ADDRESS OF $LLCTB
000536  016701  000000G                 MOV     $SLTNM,R1       ; GET NUMBER OF SYSTEM LINES
000542  006200                          ASR     R0              ; DIVIDE ADDRESSING BIAS BY 2

000544  005712                  11$:    TST     (R2)            ; IS THIS A MULTIPOINT LINE
000546  100003                          BPL     21$             ; IF PL, NO
000550  160012                          SUB     R0,(R2)         ; BIAS ADDRESS
000552  052712  100000                  BIS     #100000,(R2)    ; MAKE SURE SIGN BIT SET
000556  005722                  21$:    TST     (R2)+           ; POINT AT NEXT ENTRY
000560  005301                          DEC     R1
000562  003370                          BGT     11$             ;  AND LOOP TILL DONE
                                ;
                                ; VERIFY THAT IMAGE IS WHAT IS EXPECTED
                                ;
000564  016700  000000G                 MOV     $SLTMA,R0       ; POINT AT SLTMB
000570  016701  000000G                 MOV     $SLTNM,R1       ; NUMBER OF SLT ENTRIES
000574  022067  000000G         20$:    CMP     (R0)+,$SLTMA    ; SLT'S ADDRESS MUST BE > #$SLTMA
000600  101522                          BLOS    120$            ; IF LOS, NO - INCONSISTENT
000602  005301                          DEC     R1
000604  001373                          BNE     20$             ;  LOOP ON ALL SLTS
000606  016700  000000G                 MOV     $PDVTA,R0       ; ADDRESS OF PDV ADDRESS TABLE
000612  016701  000000G                 MOV     $PDVNM,R1       ; NUMBER OF PDVS IN SYSTEM
000616  022067  000000G         30$:    CMP     (R0)+,$LLCTA    ; PDV'S ADDRESS MUST BE > #$LLCTB
000622  101511                          BLOS    120$            ; WHAT ??
000624  005301                          DEC     R1
000626  001373                          BNE     30$             ; LOOP ON ALL PDVS


                                        .IF NDF R$$MPL                                          ;[MP01]
                                ;
                                ;       READ IN POOL LISTHEAD
                                ;
000630                                  SWTCHI  #$CEAVL-2
000636  016767  000000G 000000G         MOV     .CEAVL,$RADDR   ; SET ADDRESS OF CEX POOL
000644  062767  177776  000000G         ADD     #-2,$RADDR      ; ADJUST TO POINT AT ROUNDING FACTOR
```

```
VIMO -
Symbol

A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
C$$CKP=
C$$ORE=
C$$RSH=
D$$BUG=
D$$ISK=
D$$L11=
D$$YNC=
D$$YNM=
E$$XPR=
FE.CEX=
F$$_VL=
F.NRBD=
F.RSIZ=

. ABS.

Errors

*** Ass

Work
Work
Size of
Size of
Operati

Elapsed
SY:VIMO
```

```
                                                                                                 C 11
VIMO - VNP WRITE IMAGE OUT ROUT MACRO V05.03b Saturday 29-Jun-85 02:43  Page 5-3
Symbol table

A$$CHK= 000000          G$$TPP= 000000          M$$FCS= 000000          R$$SND= 000000          $LLCTA= ****** GX
A$$CPS= 000000          G$$TSS= 000000          M$$MGE= 000000          R$$11M= 000000          $NTLHB= ****** GX
A$$PRI= 000000          G$$TTK= 000000          M$$NET= 000000          SYSFDB= ****** GX       $NTLPT= ****** GX
A$$TRP= 000000          G$$WRD= 000000          M$$OVR= 000000          S$$WRG= 000000          $PDVNM= ****** GX
C$$CKP= 000000          I$$RAR= 000000          N$$ACC= 000001          S$$YSZ= 007600          $PDVTA= ****** GX
C$$ORE= 000400          I$$RDN= 000000          N$$BUF= 000001          T$$KMG= 000000          $PUTAD= ****** GX
C$$RSH= 177564          K$$CNT= 177546          N$$LDV= 000001          T$$MIN= 000000          $PUTRC= ****** GX
D$$BUG= 177514          K$$CSR= 177546          N$$MCP= 000001          V$$CTR= 001000          $RADDR= ****** GX
D$$ISK= 000000          K$$LDC= 000000          N$$MLL= 000001          X$$DBT= 000000          $RBLCK= ****** GX
D$$L11= 000001          K$$TPS= 000074          N$$MOV= 000010          $BFR  = ****** GX       $SAVAL= ****** GX
D$$YNC= 000000          LD$LP = 000000          N$$NCT= 000001          $BIGCT= ****** GX       $SLTMA= ****** GX
D$$YNM= 000000          L$$ASG= 000000          N$$PEM= 000001          $CEAVL= ****** GX       $SLTNM= ****** GX
E$$XPR= 000000          L$$DRV= 000000          P$$P45= 000000          $CETAB= ****** GX       $VIMO   000000RG
FE.CEX= ****** GX       L$$P11= 000001          P$$WRD= 000000          $CETDF= ****** GX       .CEAVL= ****** GX
F$$LVL= 000001          L$$11R= 000000          Q$$OPT= 000010          $CETLN= ****** GX       .MGMGE= ****** GX
F.NRBD= ****** GX       M$$CRB= 000124          R$$DER= 000000          $CEXND= ****** GX       .PDVTA= ****** GX
F.RSIZ= ****** GX       M$$CRX= 000000          R$$K11= 000001          $FMASK= ****** GX

. ABS.  000000    000    (RW,I,GBL,ABS,OVR)
        000502    001    (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9775 Words  ( 39 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.31
SY:VIMO.V2,[132,134]VIMO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VIMO
```

```
VINI
CONST
```

```
                                                                    D 11
VIMO        CREATED BY  MACRO  ON 29-JUN-85 AT 02:43        PAGE 1
```

```
VINI
```

```
                                                 .ENDC                                                      ;[MP01]

 001446  016701  000000G                         MOV     $CLBUF,R1        ; POINT AT NAME

                                         ; MOVE DEVICE AND/OR UIC TO ROOT FOR FUTURE REFERENCES

 001452  012704  000000G                 80$:    MOV     #$DFUIC,R4       ; STORAGE AREA FOR DEFAULT DEV & UIC
 001456  160002                                  SUB     R0,R2            ; NUMBER OF CHARACTERS TO MOVE
 001460  001403                                  BEQ     100$             ; NULL STRING
 001462  112024                          90$:    MOVB    (R0)+,(R4)+      ; MOVE STRING INTO PLACE
 001464  005302                                  DEC     R2
 001466  001375                                  BNE     90$              ; LOOP TILL DONE
 001470  105024                          100$:   CLRB    (R4)+            ; TERMINATE STRING
 001472  105767  000000G                         TSTB    $DFUIC           ; WAS UIC SPECIFIED?
 001476  001002                                  BNE     110$             ; BR IF YES
 001500                                          CALL    GTDUIC           ; GET DEFAULT UIC

                                         ; LOOK UP EXEC SYMBOLS

 001504  012700  000000G                 110$:   MOV     #$DFUIC,R0       ; USE SUPPLIED DEVICE AND UIC
 001510  012702  000000G                 120$:   MOV     #$STB,R2         ; USE STB EXTENSION
 001514                                          CALL    $AZFS            ; ASSEMBLE FILENAME
 001520  012700  003276'                         MOV     #EXSTB,R0        ; ADDRESS OF SYMBOL TABLE
 001524                                          CALL    $SYMBL           ; DEFINE SYMBOLS
 001530  103002                                  BCC     122$             ; IF SUCCESS - BRANCH                 ;[MP01]
 001532                                          CALLR   180$             ; ERROR IN $SYMBL
 001536                                  122$:
                                                 .IF DF  R$$MPL                                             ;[MP01]
                                                 TST     KDAS+SB.DEF      ; D-SPACE ENABLED ?                   ;[MP01]
                                                 BEQ     130$             ; IF NO BRANCH                        ;[MP01]

                                                 MOV     SCMOF+SB.VAL,.SCMOF ; COPY ADDRESS FOR END OF RO MEM  ;[MP01]
                                                 SWTCHI  # DBIAS          ; SET INPUT ADDRESS                   ;[MP01]
                                                 GETRV   .SCMOF,#2        ; READ D-SPACE BIAS                   ;[MP01]
                                                 SWTCHI                   ; RESTORE DEFAULT INPUT BUFFER        ;[MP01]
                                                 BIS     #DF.DAS,$DFLAG   ; ASSUME D-SPACE ENABLED ?            ;[MP01]
                                         130$:   MOV     FMASK+SB.VAL,R0  ; COPY ADDRESS                        ;[MP01]
                                                 ADD     #2,R0            ; GET ADDRESS OF $FMASK+2             ;[MP01]
                                                 GETRV   R0,#2            ; READ SYSTEM FEATURES WORD           ;[MP01]
                                                 BIT     #F2.DAS,$BFR     ; D-SPACE ENABLED ?                   ;[MP01]
                                                 BNE     140$             ; IF YES - BRANCH                     ;[MP01]
                                                 BIC     #DF.DAS,$DFLAG   ; IF NO - DISABLE D-SPACE             ;[MP01]

                                         140$:   BIT     #F2.POL,$BFR     ; SECONDARY POOL SUPPORTED ?          ;[MP01]
                                                 BEQ     150$             ; IF NO - BRANCH                      ;[MP01]
                                                 BIS     #DF.POL,$DFLAG   ; IF YES - ENABLE SEC. POOL SUPPORT   ;[MP01]
                                         150$:
                                                 MOV     SB.VAL+CTLST,.CTLST ; GET VALUE OF $CTLST             ;[MP01]
                                                 GETRV   .CTLST,#2        ;    ...                              ;[MP01]
                                                 MOV     $BFR,$CTLST      ;    ...                              ;[MP01]

                                                 .ENDC                                                      ;[MP01]

 001536  016767  002302  000000G                 MOV     SB.VAL+NTUIC,.NTUIC ; GET VALUE OF $NTUIC
 001544  022767  177777  000000G                 CMP     #-1,.NTUIC       ; IS $NTUIC DEFINED?
 001552  001521                                  BEQ     200$             ; BR IF NO
 001554                                          GETRV   .NTUIC,#2        ; READ IN NETWORK UIC
```

```
                                        ;
                                        ; ERROR TEXT
                                        ;

       004350                                   ERMSG$  <System image does not contain a 'GEN' partition.>
       004430                                   ERBLK$  ERR10
       004434                                   ERMSG$  <No part of 'GEN' partition is within system image.>
       004516                                   ERBLK$  ERR11
       004522                                   ERMSG$  <'GEN' partition spans highest block allocated.>
       004600                                   ERBLK$  ERR12
       004604                                   ERMSG$  <System image must be extended to %D blocks if the%n>
       004667                                   ERMSG$  <'GEN' partition is to be contained within the image.>
       004753                                   ERBLK$  ERR14,<$BFR+200>
```

```
                                        ;+
                                        ; $GETLB - READ IN A 32. WORD LIBRARY BLOCK
                                        ;
                                        ; INPUTS:
                                        ;       R0=LIBRARY BLOCK BIAS (MAY BE THE HOME BLOCK)
                                        ;
                                        ; OUTPUTS:
                                        ;       R1=VNP BUFFER ADDRESS
                                        ;-
 000000                                 $GETLB::
 000000  012701  000000G                        MOV     #$NTLHB,R1      ; ASSUME HOME BLOCK
 000004  026700  000000G                        CMP     $NTLPT,R0       ; CORRECT ASSUMPTION?
 000010  001436                                 BEQ     20$             ; IF EQ, YES - ALL DONE
 000012  012701  000002'                        MOV     #.LBUFF,R1      ; GET BUFFER ADDRESS
 000016  026700  000000'                        CMP     .LBIAS,R0       ; SAME BLOCK AS WE HAVE NOW?
 000022  001431                                 BEQ     20$             ; IF EQ, YES - ALL DONE
 000024  010067  000000'                        MOV     R0,.LBIAS       ; SET NEW LIBRARY BIAS
 000030                                         SWTCHI  R1              ; CHANGE DEFAULT BUFFER
 000034  005767  000000G                        TST     .MGMGE          ; MAPPED TARGET SYSTEM
 000040  100413                                 BMI     10$             ; IF MI, NO
 000042                                         GETAD   #0,#64.,R0
 000066  000407                                 BR      20$
 000070                                 10$:    GETRV   R0,#64.
 000106                                 20$:    SWTCHI                  ; RESET DEFAULT BUFFER
 000114                                         RETURN
```

```
                                      ;***
                                      ; Action routines
                                      ;***

                                      ;
                                      ; Store destination address for SHOW MOD X25-A
                                      ;
000322  012700  000002G               $ADS::  MOV     #RQB+LO.DES,R0  ; Buffer ptr,                              ;[TD001]
000326  012701  000006                        MOV     #MXDSTN,R1      ; Max. size of old-format destination      ;[TD001]
000332                                        CALL    STORID          ; Check and store name                     ;[TD001]
000336  103401                                BCS     101$            ; Branch if name too long                  ;[TD001]
000340                                        RETURN                                                             ;[TD001]
000342                                101$:   ERROR$  NAMETL                                                     ;[TD001]
                                                                                                                 ;[TD001]
                                      ;                                                                          ;[TD001]
                                      ; Store network name for SHOW MOD X25-A                                    ;[TD001]
                                      ;                                                                          ;[TD001]
000362  016701  000000G               $ANT::  MOV     .PSTCN,R1       ; Get length of network name               ;[TD001]
000366  020127  000020                        CMP     R1,#MAXID       ; Check not too long                       ;[TD001]
000372  101017                                BHI     101$                                                       ;[TD001]
000374  012700  000002G                       MOV     #RQB+LO.NET,R0  ; Get address to store name length         ;[TD001]
000400  110120                                MOVB    R1,(R0)+        ; Store network name length                ;[TD001]
000402  012701  000020                        MOV     #MAXID,R1       ; Clear name to spaces                     ;[TD001]
000406                                        CALL    SPFILL                                                     ;[TD001]
000412  012702  000002G                       MOV     #RQB+LO.NET,R2  ; Move network name into request block     ;[TD001]
000416  112201                                MOVB    (R2)+,R1                                                   ;[TD001]
000420  016700  000000G                       MOV     .PSTPT,R0                                                  ;[TD001]
000424                                        CALL    MOVE                                                       ;[TD001]
000430                                        RETURN                                                             ;[TD001]
                                                                                                                 ;[TD001]
000432                                101$:   ERROR$  NAMETL          ; Bad network name                         ;[TD001]


                                      ;
                                      ; Process event type
                                      ;
000452                                CKEVE:
000452  005700                                TST     R0              ; Is the value positive ?
000454  002403                                BLT     10$             ; If lt, no .. error
000456  020027  000077                        CMP     R0,#63.         ; Is it <= 63. ?
000462  003402                                BLE     20$             ; If le, yes .. okay
000464  000261                        10$:    SEC                     ; Else, indicate error
000466  000422                                BR      50$             ; And return

000470  010046                        20$:    MOV     R0,-(SP)        ; Save R0
000472  010001                                MOV     R0,R1           ; Copy event type
000474  006201                                ASR     R1              ; Get byte position in event list
000476  006201                                ASR     R1              ;       ...
000500  006201                                ASR     R1              ;       ...
000502  012702  000001                        MOV     #1,R2           ; Initialize bit position
000506  042700  177770                        BIC     #^C<7>,R0       ; Clear out all but low 3 bits
000512  001403                                BEQ     40$             ; If eq, must be bit zero
000514  006302                        30$:    ASL     R2              ; Shift to next bit position
000516  005300                                DEC     R0              ; Is this the right bit position ?
000520  003375                                BGT     30$             ; If gt, no .. continue
000522  062701  000004G               40$:    ADD     #RQB+LO.EVT,R1  ; Skip to current event list
000526  150211                                BISB    R2,(R1)         ; Set bit mask
000530  012600                                MOV     (SP)+,R0        ; Restore R0
000532  000241                                CLC                     ; Indicate success !
```

```
     004304                                      RETURN
 997                                     ;
 998                                     ; Host node address
 999                                     ;
1000 004306  005767  000100'             $STHOS::TST      AREAN             ; Use default area ?
1001 004312  001024                              BNE      10$               ; If no - branch
1002
1003 004314                                      SWTCHI   #$HMBUF           ; Set input buffer
1004 004322                                      GETRV    $DECPT,#D$END     ; Read decnet home block
1005 004342                                      SWTCHI                     ; Reset default input buffer
1006
1007 004350  016700  000014G                     MOV      $HMBUF+D$LNUM,R0          ; Get executor address
1008 004354  042700  000000C                     BIC      #^CARAMSK,R0      ; Get executor area
1009 004360  010067  000100'                     MOV      R0,AREAN          ; Store default area number
1010
1011 004364  005767  000102'             10$:    TST      ADDRS             ; Check lower addres limit
1012 004370  001413                              BEQ      101$              ; If illegal - branch
1013
1014 004372  022767  001777  000102'             CMP      #MXADDR,ADDRS     ; Check high address limit
1015 004400  002407                              BLT      101$              ; If illegal - branch
1016
1017 004402  016767  000102' 000040G             MOV      ADDRS,RQB+LO.HOS ; Store node address
1018 004410  056767  000100' 000040G             BIS      AREAN,RQB+LO.HOS ; Store node area
1019
1020 004416                                      RETURN
1021 004420                              101$:   ERROR$   ILLNAD            ; Illegal node address
1022
1023                                     ;
1024                                     ; Event logging node qualifier
1025                                     ;
1026 004440  032767  000000G 000000G     $STLND::BIT      #VF.ADD,$VFLAG    ; Node address specified?
1027 004446  001003                              BNE      10$               ; Br if yes
1028 004450  012702  000022G                     MOV      #LO.NNM+RQB,R2    ; Get address of node name storage
1029 004454  000407                              BR       STRNAM            ; Br to store node name
1030 004456  016767  000072' 000022G     10$:    MOV      NODID,LO.ADD+RQB ; Else store node address
1031 004464  052767  000000G 000000G             BIS      #VF.EAD,$VFLAG    ; Indicate node entity address specified
1032 004472                                      RETURN
1033
1034 004474  012700  000006              STRNAM: MOV      #6,R0             ; Length of node name
1035 004500  012701  000072'                     MOV      #NODID,R1         ; Address of temp storage for node name
1036 004504  112122                      10$:    MOVB     (R1)+,(R2)+       ; Store node name
1037 004506  005300                              DEC      R0                ; Loop till done
1038 004510  003375                              BGT      10$               ;       ...
1039 004512                                      RETURN
1040
1041 004514  000513                      FORWR:  BR       NODNAM            ; CONTINUE
1042
1043                                     ;
1044                                     ; Sink node name
1045                                     ;
1046 004516  032767  000000G 000000G     $STLSN::BIT      #VF.ADD,$VFLAG    ; Node address specified?
1047 004524  001003                              BNE      10$               ; Br if yes
1048 004526  012702  000014G                     MOV      #LO.SKN+RQB,R2    ; Get address of sink name storage
1049 004532  000760                              BR       STRNAM
1050 004534  016767  000072' 000014G     10$:    MOV      NODID,LO.SKA+RQB ; Store sink node address
1051 004542  052767  000000G 000000G             BIS      #VF.SKA,$VFLAG    ; Indicate sink node address specified
```

```
                                        ;
                                        ; MAXIMUM BLOCK SIZE
                                        ;
000212                                  PVPSZ:  CALL    CHKBLK          ; CHECK FOR LEGAL BLOCK SIZE VALUE
000216  103403                                  BCS     101$            ; BR IF ERROR
000220  010067  000010'                         MOV     R0,PVCPSZ       ; SAVE MAXIMUM BLOCK SIZE
000224                                          RETURN
                                        ;
                                        ; ERRORS
                                        ;
000226                                  101$:   EMSG$R  P8              ; ILLEGAL MAXIMUM BLOCK SIZE

                                        ;
                                        ; WINDOW SIZE
                                        ;
000234                                  PVWSZ:  CALL    CHKWND          ; CHECK FOR LEGAL WINDOW SIZE VALUE
000240  103403                                  BCS     101$            ; BR IF ILLEGAL VALUE
000242  010067  000012'                         MOV     R0,PVCWSZ       ; SAVE WINDOW SIZE
000246                                          RETURN
                                        ;
                                        ; ERRORS
                                        ;
000250                                  101$:   EMSG$R  00              ; ILLEGAL WINDOW SIZE

                                        ;
                                        ; PROCESS OWNER
                                        ;
000256  022767  000003  000000G         PVOWN:  CMP     #3,.PSTCN       ; LEGAL PROCESS OWNER?
000264  103410                                  BLO     101$            ; BR IF NO
000266  016700  000000G                         MOV     .PSTPT,R0       ; POINT TO STRING TO CONVERT TO RAD50
000272  005001                                  CLR     R1              ; PERIOD IS NOT A VALID TERMINATOR
000274                                          CALL    $CAT5           ; CONVERT TO RAD50
000300  010067  000014'                         MOV     R0,PVCOWN       ; SAVE PROCESS OWNER NAME
000304                                          RETURN
                                        ;
                                        ; ERRORS
                                        ;
000306                                  101$:   EMSG$R  N1              ; ILLEGAL PROCESS OWNER NAME

                                        ;
                                        ; FLAGS BYTE
                                        ;
000314  005767  000000G                 PVFG:   TST     .PNUMH          ; LEGAL VALUE?
000320  001011                                  BNE     101$            ; BR IF NO
000322  105767  000001G                         TSTB    .PNUMB+1        ; LEGAL VALUE?
000326  001006                                  BNE     101$            ; BR IF NO
000330  116767  000000G 000025'                 MOVB    .PNUMB,PVCFLG   ; SAVE FLAGS BYTE
000336                                          CALL    PVEND           ; CREATE PVC NAME BLOCK AND XCB
000342                                          RETURN
                                        ;
                                        ; ERRORS
                                        ;
000344                                  101$:   EMSG$R  N2              ; ILLEGAL FLAGS BYTE VALUE
```

```
VFPPVC -
Symbol t

. ABS.

DATA
$STATE
$KTAB
$KSTR
Errors 

*** Ass

Work  f
Work  f
Size of
Size of
Operati

Elapsed
SY:VFPP
```

D 2

```
VFPPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:38  Page 23-2
Symbol table

. ABS.  000106    000   (RW,I,GBL,ABS,OVR)
        002400    001   (RW,I,LCL,REL,CON)
DATA    000250    002   (RW,D,LCL,REL,CON)
$STATE  000214    003   (RW,D,LCL,REL,CON)
$KTAB   000006    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 15515 Words  ( 61 Pages)
Size of core pool: 16552 Words  ( 63 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:01:04.43
SY:VFPPVC.V2,[132,134]VFPPVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPPVC
```

```
VFPRDT -
TPARS STA
```

E 2

```
    142                                         .SBTTL  TPARS STATE TABLES
    143                                 ;
    144                                 ; TPARS STATE TABLES
    145                                 ;
    146 000052                                  ISTAT$  RDTST,RDTKW
    147
    148                                 ; RDT$DF
    149                                 ;
    150 000052                                  STATE$  RDTDF
    151 000052                                   TRAN$   %RDT$DF%,,,1,SYNERR
    152
    153 000052                                  STATE$
    154 000052                                   TRAN$   $STRNG,,RDNAM           ; DTE NAME
    155
    156 000052                                  STATE$
    157 000052                                   TRAN$   <',>
    158
    159 000052                                  STATE$
    160 000052                                   TRAN$   $DIGIT,,RDTEST          ; START OF DTE ADDRESS
    161
    162 000052                                  STATE$  RDTENX
    163 000052                                   TRAN$   $DIGIT,RDTENX,RDTENO
    164 000052                                   TRAN$   !END,$EXIT,RDEND        ; END OF DTE ADDRESS
    165                                 ;
    166                                 ; CHECK FOR END OF SOURCE LINE
    167                                 ;
    168 000052                                  STATE$  END
    169 000052                                   TRAN$   <';>,$EXIT
    170 000052                                   TRAN$   $EOS,$EXIT
    171
    172 000052                                  STATE$
```

D 4

```
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40  Page 7
Error messages

     85                                                  .SBTTL  Error messages
     86
     87 000000                                           .PSECT  DATA,D
     88                                                  .NLIST  BEX
     89
     90 000000                                           NTLER$  ,RA,10,CERR,RTSPC,CFLIN,<network name>
     91 000026                                           NTLER$  ,RB,10,CERR,RTSPC,CFLIN,<node name>
     92 000050                                           NTLER$  ,RC,10,CERR,RTSPC,CFLIN,<access information>
     93 000104                                           NTLER$  ,RX,8,CERR,RTSPC,CFLIN,<Remote Network Block allocation failure>
     94
     95                                                  .LIST   BEX
     96                                                  .EVEN
```

```
VFPRNW -
CKNET

    400
    401
    402
    403
    404
    405
    406
    407
    408
    409
    410
    4,1
    412
    413
    414
    415
    416
    417
    418
    419
    420
    421
    422
    423
    424
    425
    426
    427
    428
    429
    430
    431
    432
    433
```

E 4

```
                                          .SBTTL  CKNET   Check network name is unique
                              ;+
                              ; *** CKNET - Check network name is unique
                              ;
                              ; INPUTS
                              ;         $PSIPT - pointer to PSI homeblock
                              ;         RNETWK,- network name, length
                              ;         RNTL
                              ;
                              ; OUTPUTS
                              ;         Carry set if name not unique
                              ;         R3 - R5 preserved
                              ;-
 000406                       CKNET:
 000406                             SAVMAP                          ; Preserve current mapping
 000412  016700  000000G            MOV     $PSIPT,R0               ; Get address of network block listhead
 000416  062700  000042             ADD     #H$RNW,R0
 000422                             GETRV   R0,#2                   ; Read listhead
 000440  016700  000000G      10$:  MOV     $BFR,R0                 ; Test current link
 000444  000241                     CLC
 000446  001431                     BEQ     30$                     ; If EQ, end of list, hence unique
 000450                             CEACC$  R0                      ; Map to next entry in list
 000460                             GETAD   R0,#R$NWL               ; Read in next entry
 000476  116701  000002G            MOVB    $BFR+R$NML,R1           ; See if network name lengths match
 000502  120167  000000G            CMPB    R1,RNTL
 000506  001354                     BNE     10$                     ; If no match, must be unique
 000510  012702  000003G            MOV     #$BFR+R$NWK,R2          ; Address name in block
 000514  012700  000000G            MOV     #RNETWK,R0              ; Address proposed name
 000520  122022               20$:  CMPB    (R0)+,(R2)+             ; Compare names
 000522  001346                     BNE     10$                     ; If different, try next block
 000524                             SOB     R1,20$
 000530  000261                     SEC                             ; Names matched - error
 000532                       30$:  RESMAP                          ; Restore mapping
 000536                             RETURN
```

```
                                .TITLE  VFPX29 - CONFIG FILE ACTION ROUTINES
                                .IDENT  /V05.00/

                          ;
                          ; COPYRIGHT (C) 1981, 1982, 1983, 1985,  BY
                          ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                          ;
                          ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                          ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                          ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                          ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                          ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                          ; TRANSFERRED.
                          ;
                          ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                          ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                          ; CORPORATION.
                          ;
                          ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                          ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                          ;
                          ; MODULE DESCRIPTION:
                          ;
                          ; VNP - CONFIG FILE ACTION ROUTINES FOR X.29 SUPPORT
                          ;
                          ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                          ;
                          ; IDENT HISTORY:
                          ;
                          ; 1.00  02-MAR-82
                          ;       DECNET-11M V3.1
                          ;       DECNET-11MPLUS V1.1
                          ;
                          ; 3.00  16-APR-82
                          ;       DECNET-11M V3.1
                          ;       DECNET-11M-PLUS V1.1
                          ;
                          ; 4.00  07-NOV-83
                          ;       DECNET-11M V4.0
                          ;       DECNET-11M-PLUS V2.0
                          ;
                          ; 5.00  22-JUL-85
                          ;       DECnet-11M/S V4.2
                          ;       DECnet-11M-Plus V3.0
                          ;       DECnet-Micro/P      .0
                          ;
                          ;
```

```
        .TITLE  VFPX3P - INITIAL CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; VNP - CONFIG FILE ACTION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  02-MAR-82
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

D 8

```
VFPX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:41      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE              REFERENCES

$ERRO2  = ******  GX      11-258
$ERRO8    000004 R        #7-100       12-326
$ERRO9  = ******  GX      12-325
$ERRP7  = ******  GX      11-216
$ERRP8  = ******  GX      11-230
$ERRP9  = ******  GX      11-243
$ERR1T  = ******  GX       8-140
$EXIT   = 000000          #9-148
$FAIL   = 177777          #9-148
$GETAD  = ******  GX      12-298       12-316
$GETRV  = ******  GX      12-305
$GPRM   = ******           9-148
$LAMDA  = 000000          #9-148
$NUMBR  = 000002          #9-148
$PUTAD  = ******  GX      12-303       12-318
$PUTRC  = ******  GX      12-294       12-309
$QCHN     000006 RG       #8-124
$QX3P     000000 RG       #8-122
$RADDR  = ******  GX      *12-294     *12-298     *12-303     *12-305     *12-316
$RAD50  = 000016          #9-148
$RBLCK  = ******  GX      12-295      *12-319
$RONLY  = ******           9-148       9-148       9-148
$SAVAL  = ******  GX      12-288
$STRNG  = 000004          #9-148
$SUBXP  = 000010          #9-148
$$$FLG  = 177777          #9-148
$$$KEY  = 177777          #9-148
.PNUMB  = ******  GX      12-268       12-281
.PNUMH  = ******  GX      12-266       12-279
.TPARS  = ******  GX       8-131
```

```
VHELP -
Symbol

A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
C$$CKP=
C$$ORE=
C$$RSH=
D$$BUG=
D$$ISK=
D$$L11=
D$$YNC=
D$$YNM=
E$$XPR=
F$$LVL=
G$$TPP=
G$$TSS=
G$$TTK=
G$$WRD=
I$$RAR=
I$$RDN=
K$$CNT=
K$$CSR=
K$$LDC=
K$$TPS=
LD$LP =
LD.ALI=
LD.CEX=
LD.CIR=
LD.LIN=
LD.LOG=
LD.MOD=
LD.NOD=
LD.OBJ=
LD.PRO
LO.ADD
LO.ADR
LO.CLS
LO.CON
LO.COP
LO.COS
LO.CSR
LO.CTL
LO.DES
LO.DTE
LO.EVT
LO.GRO
LO.HOS
LO.HTM

. ABS.

Errors

*** As

Work
```

E 8

```
VHELP - HELP ROUTINE FOR VNP    MACRO V05.03b Saturday 29-Jun-85 02:42  Page 6-1
Symbol table

A$$CHK= 000000      LO.INA  000062      LS.ALI= 000016      OF$INS= 000001      OP$PAR= 000010
A$$CPS= 000000      LO.INC  000060      LS.CEX= 000001      OF$LOG= 000100      OP$PRI= 000010
A$$PRI= 000000      LO.INT  000046      LS.CIR= 000033      OF$OFF= 000002      OP$RET= 010000
A$$TRP= 000000      LO.LID  000030      LS.CX0= 010000      OF$ON = 000000      OP$RPA= 000010
C$$CKP= 000000      LO.LIN  000004      LS.CX5= 000400      OF$SMC= 000200      OP$SCO= 000001
C$$ORE= 000400      LO.LTM  000034      LS.ECH= 001000      OP$ADD= 000200      OP$SEG= 000400
C$$RSH= 177564      LO.MAC  000022      LS.FDX= 004000      OP$ALL= 000200      OP$SER= 000400
D$$BUG= 177514      LO.MDE  000024      LS.HDX= 010000      OP$CCS= 000004      OP$SIN= 000010
D$$ISK= 000000      LO.NAM  000044      LS.HLP= 000012      OP$CIE= 004000      OP$STA= 000004
D$$L11= 000001      LO.NET= 000002      LS.LIN= 000003      OP$CON= 000400      OP$TPA= 000020
D$$YNC= 000000      LO.NNM  000022      LS.LMC= 000007      OP$COP= 000001      OP$TRI= 000040
D$$YNM= 000000      LO.OFL  000012      LS.LOG= 000020      OP$COS= 000100      OP$UCS= 000020
E$$XPR= 000000      LO.ONM  000004      LS.MOD= 000036      OP$DST= 000200      OP$USE= 000002
F$$LVL= 000001      LO.OUT  000056      LS.NOD= 000014      OP$DTE= 000040      OP$VEC= 000040
G$$TPP= 000000      LO.OWN  000026      LS.NTI= 000013      OP$DUP= 000002      OP$VER= 000040
G$$TSS= 000000      LO.PAR  000010      LS.OBJ= 000022      OP$EST= 000100      OP$XAC= 000001
G$$TTK= 000000      LO.PRI  000012      LS.OPT= 000400      OP$EVE= 010000      OP$XPR= 000002
G$$WRD= 000000      LO.RET  000064      LS.PRO= 000002      OP$FIL= 000200      OP$X5S= 000004
I$$RAR= 000000      LO.RPA  000030      LS.TOP= 002000      OP$GRP= 000100      OP$X9S= 000010
I$$RDN= 000000      LO.SCR  000016      LS.TST= 000010      OP$HOS= 000001      P$$P45= 000000
K$$CNT= 177546      LO.SEG  000054      LS.UNF= 020000      OP$HTM= 000002      P$$WRD= 000000
K$$CSR= 177546      LO.SKA  000014      LS.XIT= 000011      OP$INA= 004000      QF$ACT= 000001
K$$LDC= 000000      LO.SKN  000014      LX.ALI= 000017      OP$INC= 001000      QF$ALL= 000002
K$$TPS= 000074      LO.SNM  000035      LX.CEX= 000004      OP$KDS= 002000      QF$KNO= 000004
LD$LP = 000000      LO.TPA  000020      LX.CIR= 000034      OP$KDT= 000400      QF$LOO= 000010
LD.ALI= 000030      LO.TRB  000034      LX.LIN= 000006      OP$KEV= 001000      Q$$OPT= 000010
LD.CEX= 000024      LO.TRI  000030      LX.LOG= 000021      OP$KGR= 001000      RQB   = ****** GX
LD.CIR= 000035      LO.UNT  000033      LX.NOD= 000015      OP$KNT= 010000      R$$DER= 000000
LD.LIN= 000025      LO.VCT  000014      LX.OBJ= 000023      OP$KSK= 002000      R$$K11= 000001
LD.LOG= 000031      LO.VER  000042      LX.PRO= 000005      OP$LCT= 000200      R$$SND= 000000
LD.MOD= 000037      LR.ACC  000020      L$$ASG= 000000      OP$LNE= 000040      R$$11M= 000000
LD.NOD= 000027      LR.ACO  000052      L$$DRV= 000000      OP$LOC= 000001      S$$WRG= 000000
LD.OBJ= 000032      LR.ADD  000012      L$$P11= 000001      OP$LST= 000002      S$$YSZ= 007600
LD.PRO= 000026      LR.ALI  000002      L$$11R= 000000      OP$LTM= 000001      TY$CHA= 000400
LO.ADD  000022      LR.BLK  000010      M$$CRB= 000124      OP$LVL= 000010      TY$EVE= 001000
LO.ADR  000052      LR.CTL  000004      M$$CRX= 000000      OP$MAC= 000010      TY$STA= 002000
LO.CLS  000002      LR.DES  000010      M$$FCS= 000000      OP$MCO= 000002      TY$SUM= 000000
LO.CON  000006      LR.HLP  000002      M$$MGE= 000000      OP$MDE= 000400      T$$KMG= 000000
LO.COP  000010      LR.LIN  000002      M$$NET= 000000      OP$MLI= 000004      T$$MIN= 000000
LO.COS  000020      LR.NAM  000012      M$$OVR= 000000      OP$MON= 000100      V$$CTR= 001000
LO.CSR  000010      LR.PAS  000041      N$$ACC= 000001      OP$MST= 000020      WC.CNT= 000400
LO.CTL  000032      LR.PRO  000002      N$$BUF= 000001      OP$NAM= 000004      WC.EVE= 004000
LO.DES  000002      LR.STS  000000      N$$LDV= 000001      OP$NET= 004000      WC.TRI= 002000
LO.DTE  000047      LR.TRI  000006      N$$MCP= 000001      OP$NLI= 000002      WC.UNT= 001000
LO.EVT  000004      LR.TYP  000002      N$$MLL= 000001      OP$NOD= 000020      X$$DBT= 000000
LO.GRO  000024      LR.UCB  000016      N$$MOV= 000010      OP$OUT= 002000      $HELP = ****** GX
LO.HOS  000040      LR.UIC  000020      N$$NCT= 000001      OP$OWN= 000020      $VHELP  000000RG
LO.HTM  000032      LR.UNT  000005      N$$PEM= 000001

. ABS.  000106      000     (RW,I,GBL,ABS,OVR)
        000010      001     (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
```

```
000652                                           GETRV   ,#6              ; READ IN 3 WORD LISTHEAD
000664  022767  000003  177776G                  CMP     #3,$CEAVL-2      ; CONSISTENCY CHECK
000672  001065                                   BNE     120$             ; IF NE, VERY BAD !!
                                                 .ENDC                                                   ;[MP01]
000674                                           SWTCHI                   ; SWITCH BACK TO DEFAULT BFR
                                         ;                                                                        ;[TD001]
                                         ; Set PSI characteristics flags                                          ;[TD001]
                                         ;                                                                        ;[TD001]
000702  016700  000000G                          MOV     $PSIPT,R0        ; Check if PSI present                  ;[TD001]
000706  001440                                   BEQ     950$                                                     ;[TD001]
000710  062700  000000                           ADD     #H$FLG,R0        ; Address PSI flags                     ;[TD001]
000714                                           GETRV   R0,#2            ; Read in PSI flags                     ;[TD001]
000732  005067  000000G                          CLR     $PFLAG           ; Initialise PSI characteristics word   ;[TD001]
000736  032767  000004  000000G                  BIT     #HF$HOS,$BFR     ; Check for host flag                   ;[TD001]
000744  001403                                   BEQ     940$                                                     ;[TD001]
000746  052767  000000G 000000G                  BIS     #PF.XGA,$PFLAG   ; If host, set XGA-present flag         ;[TD001]
000754  032767  000010  000000G 940$:            BIT     #HF$GWY,$BFR     ; Check for gateway flag                ;[TD001]
000762  001403                                   BEQ     942$                                                     ;[TD001]
000764  052767  000000G 000000G                  BIS     #PF.PSV,$PFLAG   ; If gateway, set PSV-present flag      ;[TD001]
000772  032767  000020  000000G 942$:            BIT     #HF$XDF,$BFR     ; Check for extended data flag          ;[TD001]
001000  001403                                   BEQ     950$                                                     ;[TD001]
001002  052767  000000G 000000G                  BIS     #PF.XDF,$PFLAG   ; If extended data, indicate in $PFLAG  ;[TD001]
001010                                  950$:    RETURN                                                           ;[TD001]

                                         ;
                                         ; ERRORS
                                         ;
001012  010567  000000G                 110$:    MOV     R5,$BFR                  ; SIZE OF CETAB IMAGE IN BYTES
001016  010467  000002G                          MOV     R4,$BFR+2                ; SIZE OF CET$$A PSECT IN BYTES
001022  022767  002776  000000G                  CMP     #2776,$BFR               ; MAX BUF SIZE IS 2776 OCTAL
001030  103414                                   BLO     130$                     ; YIKES !!
001032                                           ERRPT$  ERR1,$EROUT,FATAL
001046                                  120$:    ERRPT$  ERR2,EROUT,FATAL
001062                                  130$:    ERRPT$  ERR3,EROUT,FATAL
                                         ;
                                         ;       ERROR TEXT
                                         ;
001076                                           ERMSG$  <PSECT 'CET$$A' ( currently %O bytes ) should be set to%n>
001166                                           ERMSG$  <%O bytes to accomodate this exceptionally large CEX>
001251                                           ERMSG$  <%ndata base.>
001265                                           ERBLK$  ERR1,<$BFR+2,$BFR>
001274                                           ERMSG$  <CETAB data structures are inconsistent.>
001343                                           ERBLK$  ERR2
001346                                           ERMSG$  <The communications executive data structures are too large%n>
001442                                           ERMSG$  <for VNP's internal buffer. The buffer ( CET$$A PSECT )%n>
001532                                           ERMSG$  <can be expanded to 2000 bytes maximum ( from the current%n>
001624                                           ERMSG$  <%O bytes ), but this configuration needs %O bytes.>
001706                                           ERMSG$  <%nThis error is not recoverable without source modifications.>
002003                                           ERBLK$  ERR3,<GCLBUF+2,GCLBUF>

                                         ;
                                         ; EXTRA ERROR MESSAGE PROCESSING
                                         ;
002012                                  EROUT:
                                         ;
                                         ; USE $BFR BUFFER BECAUSE MESSAGE IS TOOOOOO BIG !
                                         ;
```

```
VIMO
SYMBOL
SYMBOL
FE.CEX
F.NRBD
F.RSIZ
R$$MPL
SYSFDB
$BFR
$BIGCT
$CEAVL
$CETAB
$CETDF
$CETLN
$CEXND
$FMASK
$LLCTA
$NTLHB
$NTLPT
$PDVNM
$PDVTA
$PUTAD
$PUTRC
$RADDR
$RBLCK
$SAVAL
$SLTMA
$SLTNM
$VIMO
.CEAVL
.MGMGE
.PDVTA
```

```
VIMO       CREATED BY  MACRO  ON 29-JUN-85 AT 02:43      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE          REFERENCES

FE.CEX  = ******  GX     5-168
F.NRBD  = ******  GX    *5-114    *5-125    *5-129    *5-141    *5-150    *5-163    *5-165
F.RSIZ  = ******  GX    *5-116    *5-126    *5-137    *5-145    *5-155    *5-157    *5-164
R$$MPL  = ******         5-162
SYSFDB  = ******  GX    *5-114    *5-116    *5-125    *5-126    *5-129    *5-137    *5-141    *5-145    *5-150
                        *5-155    *5-157    *5-163    *5-164    *5-165
$BFR    = ******  GX     5-165
$BIGCT  = ******  GX    *5-132    *5-136     5-139    *5-143
$CEAVL  = ******  GX     5-163
$CETAB  = ******  GX     5-125     5-129     5-141
$CETDF  = ******  GX     5-78
$CETLN  = ******  GX     5-120    *5-124     5-127     5-131     5-133    *5-144     5-145
$CEXND  = ******  GX     5-115
$FMASK  = ******  GX    *5-168
$LLCTA  = ******  GX     5-93     *5-110
$NTLHB  = ******  GX     5-150
$NTLPT  = ******  GX     5-155     5-157
$PDVNM  = ******  GX     5-80
$PDVTA  = ******  GX     5-79     *5-108     5-114     5-115     5-126     5-130
$PUTAD  = ******  GX     5-155
$PUTRC  = ******  GX     5-116     5-126     5-137     5-145     5-157     5-164
$RADDR  = ******  GX    *5-116    *5-126    *5-137    *5-145    *5-155    *5-157    *5-164
$RBLCK  = ******  GX    *5-155
$SAVAL  = ******  GX     5-74
$SLTMA  = ******  GX     5-86     *5-109
$SLTNM  = ******  GX     5-87      5-94
$VIMO     000000 RG     #5-71
.CEAVL  = ******  GX     5-164
.MGMGE  = ******  GX     5-152
.PDVTA  = ******  GX     5-116
```

```
    398 001574  016767  000000G 000000G          MOV     $BFR,$NTUIC      ; UPDATE ROOT SEGMENT VALUE
    399
    400                                  ; SET UP DEFAULT UIC AS NET UIC
    401
    402 001602  012700  000000G                  MOV     #$DFUIC,R0       ; POINT TO DEFAULT DEVICE/UIC STORAGE
    403 001606  012701  000000'                  MOV     #DEFDEV,R1       ; POINT TO DEFAULT DEVICE NAME
    404 001612  112120                   160$:   MOVB    (R1)+,(R0)+      ; STORE DEFAULT DEVICE NAME
    405 001614  001376                           BNE     160$             ; ...
    406 001616  005300                           DEC     R0               ; ...
    407 001620  005001                           CLR     R1               ; GET USER GROUP CODE
    408 001622  156701  000001G                  BISB    $NTUIC+1,R1      ; ...
    409 001626  005002                           CLR     R2               ; ALLOW ZERO SUPPRESSION
    410 001630                                   CALL    $CBOMG           ; PUT GROUP CODE IN UIC STRING
    411 001634  112720  000054                   MOVB    #<',>,(R0)+      ; STORE UIC SEPARATOR
    412 001640  005001                           CLR     R1               ; GET USER MEMBER CODE
    413 001642  156701  000000G                  BISB    $NTUIC,R1        ; ...
    414 001646  005002                           CLR     R2               ; ALLOW ZERO SUPPRESSION
    415 001650                                   CALL    $CBOMG           ; PUT USER MEMBER CODE IN STRING
    416 001654  112720  000135                   MOVB    #'],(R0)+        ; STORE UIC TREMINATOR
    417 001660  105010                           CLRB    (R0)             ; MAKE IT AN ASCIZ STRING
    418                                  ; LOOK UP COMEXEC SYMBOLS
    419
    420 001662  012700  000000G                  MOV     #$DFUIC,R0       ; USE DEFAULT VALUE SUPPLIED BY USER
    421
    422                                          .IF NDF R$$MPL
    423 001666  012701  000000G                  MOV     #$CEX,R1         ; COMEXEC ASCIZ
    424                                          .IFF
    425                                          MOV     #$RSX,R1         ; FILENAME ASCIZ
    426                                          .ENDC
    427
    428 001672  012702  000000G                  MOV     #$STB,R2         ; STB EXTENSION
    429 001676                                   CALL    $AZFS            ; COMPOSE FILENAME
    430 001702  012700  004046'                  MOV     #CEXSTB,R0       ; ADDRESS OF INTERNAL SYMBOL TABLE
    431 001706                                   CALL    $SYMBL           ; LOOK UP UNDEFINED SYMBOLS
    432 001712  103433                           BCS     190$             ; ERROR ?
    433
    434                                  ; NOW MOVE THE SYMBOLS TO THE ROOT SEGMENT
    435
    436 001714                                   CALL    SYMOV            ; MOVE THE SYMBOLS TO RESTING PLACE
    437 001720  016700  000000G                  MOV     .FMASK,R0        ; GET ADDRESS OF FEATURES MASK
    438 001724  062700  000004                   ADD     #4,R0            ; POINT TO $FMASK+4 (3RD FEATURES MASK)
    439 001730                                   GETRV   R0,2             ; READ IN FEATURES MASK
    440 001746  032767  000400  000000G          BIT     #F3.11S,$BFR     ; IS THIS AN 11S SYSTEM?
    441 001754  001403                           BEQ     170$             ; BR IF NO
    442 001756  052767  000000G 000000G          BIS     #VF.SS,$VFLAG    ; ELSE INDICATE 11S
    443 001764                           170$:   RETURN
    444
    445                                  ;
    446                                  ; ERRORS
    447                                  ;
    448
    449 001766                           180$:   ERRPT$  ERR4,$EROUT,FATAL
    450 002002                           190$:   ERRPT$  ERR5,$EROUT,FATAL
    451 002016  016167  000004  000000G  200$:   MOV     SB.SYB(R1),$BFR ; LOAD UNDEFINED SYMBOL
    452 002024  016167  000006  000002G          MOV     SB.SYB+2(R1),$BFR+2 ;   RAD50 NAME
    453 002032                                   ERRPT$  ERR6,$EROUT,FATAL
    454                                  ;
```

```
   1009                                         .SBTTL  GTDUIC - GET DEFAULT UIC
   1010
   1011                                 ;+
   1012                                 ;
   1013                                 ; INPUTS:
   1014                                 ;       NONE
   1015                                 ;
   1016                                 ; OUTPUTS:
   1017                                 ;       $DFUIC CONTAINS DEFAULT UIC
   1018                                 ;
   1019                                 ;-
   1020 004760                          GTDUIC: CALL    $SAVAL                  ; SAVE ALL REGISTERS
   1021 004764                                  CALL    .RDFUI                  ; GET DEFAULT UIC
   1022 004770  010103                          MOV     R1,R3                   ; GET BINARY UIC VALUE
   1023 004772  012702  000004G                 MOV     #$DFUIC+4,R2            ; POINT TO BEGINNING OF UIC STRING
   1024 004776  012704  000001                  MOV     #UICCFG,R4              ; SET UP UIC CONVERSION FLAG
   1025 005002                                  CALL    .PPASC                  ; CONVERT TO ASCII STRING
   1026 005006                                  RETURN
   1027
   1028         000001                          .END
```

```
106                                     ;+
107                                     ; $PUTLB - RE-WRITE 32. WORD LIBRARY BLOCK
108                                     ;
109                                     ; INPUTS:
110                                     ;       R0=LIBRARY BLOCK BIAS (MAY BE THE HOME BLOCK)
111                                     ;
112                                     ; OUTPUTS:
113                                     ;       NONE
114                                     ;-
115 000116                              $PUTLB::
116 000116  026700  000000G                     CMP     $NTLPT,R0               ; IS IT THE HOME BLOCK?
117 000122  001432                              BEQ     20$                     ; IF EQ, YES - ALL DONE
118 000124                                      SWTCHO  #.LBUFF                 ; CHANGE DEFAULT BUFFER
119 000132  005767  000000G                     TST     .MGMGE                  ; MAPPED TARGET SYSTEM?
120 000136  100414                              BMI     10$                     ; IF MI, NO
121 000140                                      PUTAD   #0,#64.,.LBIAS
122 000166  000410                              BR      20$
123 000170                              10$:    PUTRC   .LBIAS,#64.
124 000210                              20$:    SWTCHO                          ; RESET DEFAULT BUFFER
125 000216                                      RETURN
126
127         000001                              .END
```

```
000534                           50$:    RETURN


                                 ;
                                 ; Check event range beginning
                                 ;
000536                           CKEVEB::
000536  005767   000000G                 TST     .PNUMH          ; Is it a word value ?
000542  001003                           BNE     10$             ; If ne, no .. error
000544  016700   000000G                 MOV     .PNUMB,R0       ; Get beginning event type
000550  002003                           BGE     20$             ; If non-zero .. okay
000552  062716   000002          10$:    ADD     #2,(SP)         ; Else, reject
000556  000402                           BR      30$             ; And return

000560  010067   000062'         20$:    MOV     R0,EVEBEG       ; Save beginning event type
000564                           30$:    RETURN

                                 ;
                                 ; Check event class
                                 ;

000566                           CKEVEC::
000566  005767   000000G                 TST     .PNUMH          ; Is the event class a word value ?
000572  001006                           BNE     10$             ; If ne, no .. error
000574  016700   000000G                 MOV     .PNUMB,R0       ; Else, get event class number
000600  002403                           BLT     10$             ; If eq, error
000602  020027   000777                  CMP     R0,#511.        ; Is the number in range ?
000606  003403                           BLE     20$             ; If le, yes .. okay
000610  062716   000002          10$:    ADD     #2,(SP)         ; Else, reject transition
000614  000410                           BR      30$             ; And return

000616  000006                   20$:    .REPT   6
                                         ASL     R0              ; Get event class in bits 6-15
                                         .ENDR
000632  010067   000002G                 MOV     R0,RQB+LO.CLS   ; And store the event class
000636                           30$:    RETURN

                                 ;
                                 ; Check event number
                                 ;
000640                           CKEVEN::
000640  005767   000000G                 TST     .PNUMH          ; Is the event type a word value ?
000644  001005                           BNE     10$             ; If ne, no .. error
000646  016700   000000G                 MOV     .PNUMB,R0       ; Else, get the event type
000652                                   CALL    CKEVE           ; And check the type value
000656  103002                           BCC     20$             ; If cc, all okay
000660  062716   000002          10$:    ADD     #2,(SP)         ; Else, reject
000664                           20$:    RETURN

                                 ;
                                 ; Check event range
                                 ;
000666                           CKEVER::
000666  016700   000062'                 MOV     EVEBEG,R0       ; Get beginning event type
000672  005767   000000G                 TST     .PNUMH          ; Is this a word value ?
000676  001003                           BNE     10$             ; If ne, no .. error
000700  026700   000000G                 CMP     .PNUMB,R0       ; Is value > beginning value ?
000704  003003                           BGT     20$             ; If gt, yes .. okay
```

```
004550                                  RETURN

004552  012767  177777  000014G $STLSH::MOV     #-1,LO.SKA+RQB  ; SET $HOST AS SINK NODE
004560  052767  000000G 000000G         BIS     #VF.SKA,$VFLAG  ; INDICATE SINK NODE ADDRESS SPECIFIED
004566                                  RETURN


                                ;
                                ; Store node address for set node.
                                ;
004570  005767  000100'         $STNAD::TST     AREAN           ; Use default area ?
004574  001024                          BNE      0$             ; If no - branch

004576                                  SWTCHI  #$HMBUF         ; Set up input buffer
004604                                  GETRV   $DECPT,#D$END   ; Read decnet home block
004624                                  SWTCHI                  ; Reset input buffer

004632  016700  000014G                 MOV     $HMBUF+D$LNUM,R0 ; Get executor address
004636  042700  000000C                 BIC     #^CARAMSK,R0    ; Get area number
004642  010067  000100'                 MOV     R0,AREAN        ; Store default area number

004646  005767  000102'         10$:    TST     ADDRS           ; Check lower address limit
004652  001413                          BEQ     101$            ; If illegal - branch

004654  022767  001777  000102'         CMP     #MXADDR,ADDRS   ; Check high address limit
004662  002407                          BLT     101$            ; If illegal - branch

004664  016767  000102' 000022G         MOV     ADDRS,RQB+LO.ADD ; Set node address
004672  056767  000100' 000022G         BIS     AREAN,RQB+LO.ADD ; Set area number

004700                                  RETURN

004702                          101$:   ERROR$  ILLNAD          ; Illegal node address.


                                ;
                                ; Node name
                                ;
004722                          $NNAM::
004722  012702  000072'                 MOV     #NODID,R2       ; Get address of output buffer
004726  012701  000006                  MOV     #MXNODN,R1      ; Buffer size.
004732  000404                          BR      NODNAM          ; And continue in common code
                                ;
                                ; Remote node name
                                ;
004734                          $STNAM::
004734  012702  000044G                 MOV     #RQB+LO.NAM,R2  ; Get address of output buffer
004740  012701  000006                  MOV     #MXNODN,R1      ; Buffer size.

004744                          NODNAM:                         ; Store node name
004744  016700  000000G                 MOV     .PSTPT,R0       ; Get start of node name
004750  020167  000000G                 CMP     R1,.PSTCN       ; Is node name too long ?
004754  103414                          BLO     101$            ; If lo, yes
004756  010146                          MOV     R1,-(SP)        ; Save node name field length

004760  112722  000040          10$:    MOVB    #40,(R2)+       ; Blank fill node name
004764  005301                          DEC     R1              ;       ...
004766  003374                          BGT     10$             ;       ...

```

```

                                        ;                                                                               ;[TD001]
                                        ; End of PVC$DF macro - allocate PVC name block                                ;[TD001]
                                        ;                                                                               ;[TD001]
 000352                                 PVEND:                                                                          ;[TD001]
 000352                                         CALL    FNDPRT          ; Look for a free port entry                    ;[TD001]
 000356  103425                                 BCS     102$            ; Branch if none avaiable                       ;[TD001]
 000360  032767  000000G 000000G                BIT     #PF.XDF,$PFLAG  ; Check if extended data format                 ;[TD001]
 000366  001407                                 BEQ     10$             ; Branch if not                                 ;[TD001]
                                        ;                                                                               ;[TD001]
                                        ; Allocate PVC name block in extended format                                    ;[TD001]
                                        ;                                                                               ;[TD001]
 000370                                         CALL    CKPVC0          ; Check for valid PVC id                        ;[TD001]
 000374  103413                                 BCS     101$            ; Branch if invalid                             ;[TD001]
 000376                                         CALL    ALPVC0          ; Allocate and fill PVC name block              ;[TD001]
 000402  103413                                 BCS     102$            ; Branch if no store                            ;[TD001]
 000404  000406                                 BR      20$                                                             ;[TD001]
                                        ;                                                                               ;[TD001]
                                        ; Allocate PVC name block in old format                                         ;[TD001]
                                        ;                                                                               ;[TD001]
 000406                                 10$:    CALL    CKPVC1          ; Check for valid PVC id                        ;[TD001]
 000412  103404                                 BCS     101$            ; Branch if invalid                             ;[TD001]
 000414                                         CALL    ALPVC1          ; Allocate and fill PVC name block              ;[TD001]
 000420  103404                                 BCS     102$            ; Branch if no store                            ;[TD001]
                                        ;                                                                               ;[TD001]
 000422                                 20$.                                                                            ;[TD001]
 000422                                         RETURN                                                                  ;**-34
                                        ;
                                        ; ERRORS
                                        ;
 000424                                 101$:   EMSG$R  N0              ; INVALID PVC ID
 000432                                 102$:   EMSG$R  N3              ; PVC/XCB/PORT/HASH ALLOCATION FAILED
```

```
VFPPVC

SYMBOL

SYMBOL

ALLXCB
ALPVC0
ALPVC1
BIASX
BLKSZD
CERR
CFGBF
CFGSZ
CFLIN
CHKBLK
CHKWND
CHNLMX
CKPVC0
CKPVC1
C$DTE
C$FLG
C$GNAM
C$LCN
C$LEN
C$NAM
C$NML
C$PORT
DLMCTL
DLMIND
DLMPDV
DLMSLT
DLMSTA
DLMUNT
DTEDES
FLAG
FMT10
FMT8
FM.10
FM.8
FNDHSH
FNDPDV
FNDPRT
F.RSIZ

HSHADD
HSHSZ
H$PTB
H$PVC
INIXCB
INVCIR
LNKEND
MAXID
MOVE
N$$VCT
PCVKW
PF.DLM
PF.PSV
```

```
VFPPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:38      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE              REFERENCES

ALLXCB     000440 R         #14-427    17-574     18-622
ALPVC0     001152 R          13-393   #17-548
ALPVC1     001344 R          13-401   #18-600
BIASX      002256 R          20-706   #22-785
BLKSZD   = ******   GX       11-286
CERR     = ******   GX       8-145      8-146      8-147      8-148
CFGBF    = ******   GX       9-173
CFGSZ    = ******   GX       9-172
CFLIN    = ******   GX       8-145      8-146      8-147      8-148
CHKBLK   = ******   GX       12-328
CHKWND   = ******   GX       12-340
CHNLMX   = ******   GX       11-303
CKPVC0     000664 R          13-391   #15-475
CKPVC1     001022 R          13-399   #16-513
C$DTE      000012           *17-566   *18-616
C$FLG      000015           *17-564   *18-614
C$GNAM     000016            15-490    17-559
C$LCN      000010           *17-565   *18-615
C$LEN      000016            15-485    16-523     17-553     17-569     18-604     18-618
C$NAM      000002            16-526    18-610
C$NML      000002            15-487   *17-562
C$PORT     000014           *17-567   *18-617
DLMCTL     000047 R         #7-131    *23-824
DLMIND     000052 R         #7-134    *23-813
DLMPDV     000016 R         #7-123    *23-814
DLMSLT     000020 R         #7-124
DLMSTA     000051 R         #7-133    *23-826
DLMUNT     000050 R         #7-132
DTEDES   = ******   GX       17-566    18-616     20-710     21-745
FLAG       000022 R         #7-125    *9-168
FMT10    = ******   GX       8-145     8-146      8-147
FMT8     = ******   GX       8-148
FM.10    = 000000           #8-145    #8-146     #8-147
FM.8     = 000000           #8-148
FNDHSH     002120 R          14-446   #21-744
FNDPDV   = ******   GX       14-455    23-811
FNDPRT     001522 R          13-384   #19-651
F.RSIZ   = ******   GX      *14-435   *14-441    *14-452    *15-479    *15-485    *16-517    *16-523    *17-570    *18-618
                            *19-655   *19-659    *20-700    *21-757
HSHADD   = ******   GX       21-756
HSHSZ    = ******   GX       21-750    21-761
H$PTB      000020            14-434    19-654
H$PVC      000006            15-478    16-516     17-580     18-628
INIXCB     001676 R          14-445   #20-695
INVCLR     002336 R          23-808    23-812    #23-819     23-828     23-830
LNKEND   = ******   GX       17-581    18-629
MAXID    = 000020           #6-101     7-130      15-485
MOVE     = ******   GX       17-563    18-613
N$$VCT   = ******            15-484    16-522     17-558     18-609     19-656     20-695
PCVKW      000000 RG         9-171    #10-190
PF.DLM   = 000001           #7-126
PF.PSV   = ******   GX       14-428    14-449     20-697
```

```
VFPRDT -
RDT$DF A

   174
   175
   176
   177
   178
   179
   180
   181
   182
   183
   184
   185
   186
   187
   188
   189
   190
   191
   192
   193
   194
   195
   196
   197
   198
   199
   200
   201
   202
   203
   204
   205
   206
   207
   208
   209
   210
   211
   212
   213
   214
   215
   216
   217
   218
   219
   220
   221
```

```
    174                                                     .SBTTL  RDT$DF ACTION ROUTINES
    175
    176                                             ;
    177                                             ; REMOTE DTE NAME
    178                                             ;
    179 000052  016700  000000G                     RDNAM:  MOV     .PSTCN,R0               ; GET NUMBER OF CHARACTERS SPECIFIED
    180 000056  012702  000006                              MOV     #RDTEMX,R2              ; GET MAXIMUM ALLOWED
    181 000062  020002                                      CMP     R0,R2                   ; LEGAL NAME LENGTH?
    182 000064  101016                                      BHI     101$                    ; BR IF NO
    183 000066  012701  000000'                             MOV     #RDTNAM,R1              ; POINT TO REMOTE DTE NAME STORAGE
    184 000072  112721  000040                      5$:     MOVB    #SPACE,(R1)+            ; FILL WITH SPACES
    185 000076  005302                                      DEC     R2                      ; MORE TO PAD?
    186 000100  003374                                      BGT     5$                      ; BR IF YES
    187 000102  012701  000000'                             MOV     #RDTNAM,R1              ; POINT TO REMOTE DTE NAME STORAGE
    188 000106  016702  000000G                             MOV     .PSTPT,R2               ; POINT TO SPECIFIED NAME
    189 000112  112221                              10$:    MOVB    (R2)+,(R1)+             ; STORE SPECIFIED REMOTE DTE NAME
    190 000114  005300                                      DEC     R0                      ; MORE TO STORE?
    191 000116  003375                                      BGT     10$                     ; BR IF YES
    192 000120                                              RETURN
    193                                             ;
    194                                             ; ERRORS
    195                                             ;
    196 000122                                      101$:   EMSG$R  L0                      ; ILLEGAL REMOTE DTE NAME
    197
    198                                             ;
    199                                             ; START OF REMOTE DTE ADDRESS
    200                                             ;
    201 000130  012700  000017                      RDTEST: MOV     #RDTLN,R0               ; GET LENGTH OF REMOTE DTE ADDRESS
    202 000134  012701  000006'                             MOV     #RDTE,R1                ; GET ADDRESS OF BUFFER
    203 000140  105021                              10$:    CLRB    (R1)+                   ; INITIALIZE BUFFER
    204 000142  005300                                      DEC     R0                      ; MORE TO INITIALIZE?
    205 000144  003375                                      BGT     10$                     ; BR IF YES
    206 000146  105067  000000G                             CLRB    COUNT                   ; INITIALIZE DIGIT COUNT
    207 000152  012767  000006' 000000G                     MOV     #RDTE,NEXT              ; START STORING AT BEGINNING OF BUFFER
    208 000160                                              CALL    STRNXT                  ; STORE CHARACTER
    209 000164                                              RETURN
    210
    211                                             ;
    212                                             ; PROCESS REMOTE DTE DIGIT
    213                                             ;
    214 000166  026727  000000G 000024'             RDTENO: CMP     NEXT,#RDTEN             ; TOO MANY DIGITS?
    215 000174  101                                         BHI     101$                    ; BR IF YES
    216 000176                                              CALL    STRNXT                  ; STORE NEXT DIGIT
    217 000202                                              RETURN
    218                                             ;
    219                                             ; ERRORS
    220                                             ;
    221 000204                                      101$:   EMSG$R  L1                      ; ILLEGAL REMOTE DTE ADDRESS
```

```
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40  Page 8
Local Data

     98                                          .SBTTL  Local Data
     99
    100 000164                          STRLEN: .BLKB   1                       ; Length of parsed string
    101 000165                          STRBUF: .BLKB   MAXSTR                  ; Buffer for parsed string
    102
    103                                         .EVEN
    104
    105 000000                                  .PSECT
```

```
VFPRNW -
ALNET

   435
   436
   437
   438
   439
   440
   441
   442
   443
   444
   445
   446
   447
   448
   449
   450
   451
   452
   453
   454
   455
   456
   457
   458
   459
   460
   461
   462
   463
   464
   465
   466
   467
   468
   469
   470
   471
   472
   473
   474
   475
   476
   477
   478
   479
   480
   481
   482
   483
   484
   485
   486
   487
   488
   489
   490
   491
```

```
                                        .SBTTL  ALNET   Allocate remote network block
                                ;+
                                ; *** ALNET - Allocate remote network block
                                ;
                                ; INPUTS
                                ;       All parsed fields for RNW$DF/RNA$DF
                                ;       $PSIPT - pointer to PSI homeblock
                                ;
                                ; OUTPUTS
                                ;       Carry clear - network block allocated and linked
                                ;       Carry set - no space for block
                                ;       All registers preserved
                                ;
                                ;-
000540                          ALNET:
000540                                  CALL    $SAVAL                  ; Save all registers
000544                                  SAVMAP                          ; Save current mapping
000550  012701  000104                  MOV     #R$NWL,R1               ; Length of block to allocate
000554                                  CALL    $ALPOL                  ; Attempt allocation
000560  103511                          BCS     100$                    ; Branch if failure
000562  010005                          MOV     R0,R5                   ; Save unmapped address of allocation
000564                                  CEACC$  R0,R4                   ; Map to allocation
000574  012700  000000G                 MOV     #$BFR,R0                ; Clear allocated area to zeroes
000600  012701  000104                  MOV     #R$NWL,R1
000604  105020                  10$:    CLRB    (R0)+
000606                                  SOB     R1,10$
                                ;
000612  012700  000000G                 MOV     #RNTL,R0                ; Get address of network name buffer
000616  112001                          MOVB    (R0)+,R1                ; Get name length
000620  110167  000002G                 MOVB    R1,$BFR+R$NML           ; Insert length into block
000624  012702  000003G                 MOV     #$BFR+R$NWK,R2          ; Target for network name
000630                                  CALL    MOVE                    ; Insert network name into block
000634  012700  000023G                 MOV     #$BFR+R$NOD,R0          ; Target for node name
000640  012701  000006                  MOV     #MXNOID,R1
000644                                  CALL    SPFILL                  ; Space fill node name
000650  010002                          MOV     R0,R2                   ; Target for node name
000652  012700  000000G                 MOV     #RNNL,R0                ; Address of node name buffer
000656  112001                          MOVB    (R0)+,R1                ; Length of node name
000660                                  CALL    MOVE                    ; Insert node name into block
000664  012700  000000G                 MOV     #RNUL,R0                ; Get address of user name buffer
000670  112001                          MOVB    (R0)+,R1                ; Get user name length
000672  110167  000031G                 MOVB    R1,$BFR+R$USL           ; Insert user name length into block
000676  012702  000032G                 MOV     #$BFR+R$USR,R2          ; Target for user name
000702                                  CALL    MOVE                    ; Insert user name into block
000706  012700  000000G                 MOV     #RNPL,R0                ; Get address of password buffer
000712  112001                          MOVB    (R0)+,R1                ; Get length of password
000714  110167  000052G                 MOVB    R1,$BFR+R$PAL           ; Insert password length into buffer
000720  012702  000053G                 MOV     #$BFR+R$PAS,R2          ; Target for password
000724                                  CALL    MOVE                    ; Insert password into buffer
000730  012700  000000G                 MOV     #RNAL,R0                ; Get address of account buffer
000734  112001                          MOVB    (R0)+,R1                ; Get length of account
000736  110167  000063G                 MOVB    R1,$BFR+R$ACL           ; Insert account length into block
000742  012702  000064G                 MOV     #$BFR+R$ACC,R2          ; Target for account
000746                                  CALL    MOVE                    ; Insert account into buffer
000752                                  PUTAD   R4,#R$NWL               ; Write out block to system image
000770  016700  000000G                 MOV     $PSIPT,R0               ; Get address of listhead
000774  062700  000042                  ADD     #H$RNW,R0
001000                                  CALL    LNKEND                  ; Link block (unmapped addr in R5)
```

```
     52                                                        .SBTTL  MACRO DEFINITIONS
     53
     54                                                ;
     55                                                ; LIBRARY MACROS
     56                                                ;
     57                                                        .MCALL  EMSG$R,NTLER$,X29DF$,PHBDF$,ISTAT$,STATE$,TRAN$
     58                                                        .MCALL  GETRV,PUTRC
     59
     60 000000                                                 X29DF$                          ; X.29 DATA BLOCK DEFINITIONS
     61 000000                                                 PHBDF$                          ; PSI HOME BLOCK DEFINTIONS
     62
```

```
VFPX3P - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:41  Page 5
MACRO DEFINITIONS

     55                                          .SBTTL  MACRO DEFINITIONS
     56
     57                                  ;
     58                                  ; LIBRARY MACROS
     59                                  ;
     60                                          .MCALL  EMSG$R,CHNDF$,DTEDF$,NTLER$,TRAN$,STATE$,ISTAT$
     61                                          .MCALL  GETRV,PUTRC,GETAD,PUTAD,CEACC$
     62
     63 000000                                   CHNDF$                  ; DEFINE CHANNEL TABLE OFFSETS
     64 000000                                   DTEDF$                  ; DEFINE DTE DESCRIPTOR OFFSETS
     65
     66                                  ;
     67                                  ; LOCAL MACRO DEFINITIONS
     68                                  ;
     69                                          .MACRO  SAVMAP
     70                                          MOV     $RBLCK,-(SP)
     71                                          .ENDM
     72
     73                                          .MACRO  RESMAP
     74                                          MOV     (SP)+,$RBLCK
     75                                          .ENDM
     76
```

```
VFPX3

MACRO

MACRO

CALL

CEAC
CHND
DBGT
DTED
EMSG
GETA
GETR
ISTA
MTRA
NTLE
PUTA
PUTR
RAND

RESM
RETU
SAVM
STAT

TRAN
```

```
VFPX3P     CREATED BY  MACRO  ON 29-JUN-85 AT 02:41     PAGE 3
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME      REFERENCES

CALL              8-131     11-209    11-221    11-236    11-248    12-288    12-289    12-294    12-296    12-298
                  12-303    12-305    12-309    12-316    12-318
CEACC$           #5-61      12-296
CHNDF$           #5-60      5-63
DBGTP$           #9-148    #10-190   #10-200
DTEDF$           #5-60      5-64
EMSG$R           #5-60      8-140     11-216    11-230    11-231    11-243    11-257    11-258    12-325    12-326
GETAD            #5-61      12-298    12-316
GETRV            #5-61      12-305
ISTAT$           #5-60      9-148
MTRAN$           #9-148
NTLER$           #5-60      7-100
PUTAD            #5-61      12-303    12-318
PUTRC            #5-61      12-294    12-309
RAND             #12-294    12-294   #12-298    12-298   #12-303    12-305   #12-305    12-305   #12-309    12-309
                 #12-316    12-316   #12-318    12-318
RESMAP           #5-73      12-319
RETURN            8-135     11-212    11-226    11-239    11-253    12-274    12-320
SAVMAP           #5-69      12-295
STATE$           #5-60      9-153    #9-156    #9-159    #9-162    #9-165    #9-168    #9-171    #9-174     10-182
                 #10-185   #10-188   #10-192   #10-198   #10-202
TRAN$            #5-60     #9-154    #9-157    #9-160    #9-163    #9-166    #9-169    #9-172    #9-175    #10-183
                 #10-186   #10-189   #10-190   #10-193   #10-199   #10-200
```

```
VHELP -
Symbol

Work
Size o
Size o
Operat

Elapse
SY:VHEL
```

```
VHELP - HELP ROUTINE FOR VNP    MACRO V05.03b Saturday 29-Jun-85 02:42  Page 6-2
Symbol table

Work  file writes: 0
Size of work file: 12930 Words  ( 51 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:08.28
SY:VHELP.V2,[132,134]VHELP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VHELP
```

```
    341 002012  012767  000000G 000000G         MOV     #$BFR,$CLBUF         ; SET NEW GCL BUFFER
    342 002020  016767  000000G 000000G         MOV     $BFR,GCLBUF          ; GET PARAMETER VALUES
    343 002026  016767  000002G 000002G         MOV     $BFR+2,GCLBUF+2      ; ...
    344 002034  000167  000000G                 JMP     $EROUT               ; PRINT OUT MESSAGE
    345
    346
    347         000001                          .END
```

```
VIMO

MACRO (

MACRO N

CALL
ERBLK$
ERMSG$
ERRPT$
GETRV
PUTAD
PUTRC
RAND

RETURN
SWTCHO
VNPBPT
```

```
VIMO        CREATED BY  MACRO  ON 29-JUN-85 AT 02:43     PAGE 2
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME       REFERENCES

CALL               5-74      5-116      5-126      5-137      5-145      5-155      5-157      5-164
ERBLK$            #5-73
ERMSG$            #5-73
ERRPT$            #5-73
GETRV             #5-68
PUTAD             #5-68      5-155
PUTRC             #5-68      5-116      5-126      5-137      5-145      5-157      5-164
RAND              #5-116     5-116     #5-126      5-126     #5-137      5-137     #5-145      5-145     #5-155      5-155
                  #5-157     5-157     #5-164      5-164
RETURN             5-169
SWTCHO            #5-68      5-114      5-125      5-129      5-141      5-150      5-163      5-165
VNPBPT            #5-72      5-73
```

```
    455                                  ; error text
    456                                  ;
    457
    458 002046                                   ERMSG$  <Executive symbol table file open failure ( %E )>
    459 002125                                   ERMSG$  <%n      ( %A )>
    460 002136                                   ERBLK$  ERR4,<$CLFDB+F.ERR,FILE>
    461 002146                                   ERMSG$  <Communications executive symbol table file open%n>
    462 002227                                   ERMSG$  <failure ( %E ) ( %A ).>
    463 002255                                   ERBLK$  ERR5,<$CLFDB+F.ERR,FILE>
```

```
VINI
Symbo

ABTIM
AS.DE
AS.EX
AS.RE
AS.WR
A$$CH
A$$CR
A$$PR
A$$TR
A.IOC
A.LG
A.MP
A.PCB
A.PCB
A.PR
A.STA
A.TCB
A.TCB
CEAV
CEPW
CEXS
CLIN
CLKH
CL.A
CL.A
CL.C
CL.C
CL.E
CL.E
CL.E
CL.I
CL.L
CL.N
CL.O
CL.R
CL.X
CMOD
CRAV
CRCO
CS.D
CS.D
CS.E
CS.I
CS.M
CS.N
CS.O
CS.W
C$$C
C$$C
C$$R
C.CM
C.DE
C.DI
C.DS
C.FI
C.MK
C.MK
```

```
                                                                      E 13
VINI - VNP INITIALIZATION       MACRO V05.03b Saturday 29-Jun-85 02:43  Page 14-1
Symbol table

ABTIM   003726R      C.SIZE= 000054       F.ACTL= 000076       F2.LIB= 000002       LD$LP = 000000
AS.DEL= 000010       C.STAT= 000001       F.ALOC= 000040       F2.MP = 000004       LD$REL= 000004
AS.FXT= 000004       C.SWAD= 000022       F.BBFS= 000062       F2.POL= 000100       LD$RSV= 040000
AS.RED= 000001       C.TYPR= 000000       F.BDB = 000070       F2.RAS= 004000       LD$SUP= 000010
AS.WRT= 000002       DEFDEV  000000R      F.BGBC= 000057       F2.RBN= 020000       L$BASG  001000
A$$CHK= 000000       DEUMR   003702R      F.BKDN= 000026       F2.SDW= 000040       L$BBLK  000360
A$$CPS= 000000       DEVHD   003334R      F.BKDS= 000020       F2.STP= 100000       L$BDAT  000032
A$$PRI= 000000       DLCOV   004072R      F.BKEF= 000050       F2.SWP= 040000       L$BEXT  000352
A$$TRP= 000000       D$$BUG= 177514       F.BKP1= 000051       F2.WND= 000200       L$BFLG  000030
A.IOC   000003       D$$ISK= 000000       F.BKST= 000024       F3.AST= 000200       L$BFL2  000774
A.LGTH= 000014       D$$L11= 000001       F.BKVB= 000064       F3.CLI= 001000       L$BHGV  000012
A.MPCT  000011       D$$YNC= 000000       F.CHR = 000075       F3.CRA= 000001       L$BHRB  000356
A.PCB   000012       D$$YNM= 000000       F.CNTG= 000034       F3.EIS= 000004       L$BLDZ  000016
A.PCBL  000000       ERROR1  000200R      F.DFNB= 000046       F3.NWK= 000002       L$BLIB  000040
A.PRI   000002       ERR0    000724R      F.DSPT= 000044       F3.PMN= 004000       L$BLRL  000776
A.STAT  000010       ERR1    000770R      F.DVNM= 000134       F3.PRO= 000040       L$BLUN  000362
A.TCB   000004       ERR10   004432R      F.EFBK= 000010       F3.RLK= 020000       L$BMXV  000014
A.TCBL  000006       ERR11   004520R      F.EFN = 000050       F3.SHF= 040000       L$BMXZ  000020
CEAVL   004060R      ERR12   004602R      F.EOBB= 000032       F3.STM= 000010       L$BOFF  000022
CEPWR   004116R      ERR13   001310R      F.ERR = 000052       F3.TCM= 002000       L$BPAR  000004
CEXSTB  004046R      ERR14   004754R      F.EXT = 000072       F3.UDS= 000020       L$BPRI  000346
CLINS   003430R      ERR2    001060R      F.FACC= 000043       F3.WAT= 010000       L$BROB  000364
CLKHD   003752R      ERR3    001170R      F.FFBY= 000014       F3.XHR= 000100       L$BROL  000366
CL.AST= 004000       ERR4    002140R      F.FLG = 000074       F3.11S= 000400       L$BSA   000010
CL.ATT= 000200       ERR5    002256R      F.FNAM= 000110       GTDUIC  004760R      L$BSEG  000026
CL.CAS= 020000       ERR6    002536R      F.FNB = 000102       G$$TPP= 000000       L$BSGL  000354
CL.CMT= 000001       ERR7    002626R      F.FTYP= 000116       G$$TSS= 000000       L$BSYS  000025
CL.EIF= 000040       EXSIZ   003346R      F.FVER= 000120       G$$TTK= 000000       L$BTSK  000000
CL.EXN= 040000       EXSTB   003276R      F.HIBK= 000004       G$$WRD= 000000       L$BWND  000024
CL.EXT= 000010       E$$XPR= 000000       F.LUN = 000042       HEADR   003454R      L$BXFR  000350
CL.IND= 000002       FATAL = ******  GX   F.MBCT= 000054       HF.CIS= 000200       L$$ASG= 000000
CL.LCC= 000020       FD.INS= ******  GX   F.MBC1= 000055       HF.EIS= 000002       L$$DRV= 000000
CL.NPR= 000400       FE.CAL= 000040       F.MBFG= 000056       HF.FPP= 100000       L$$P11= 000001
CL.OPN= 000100       FE.CEX= 020000       F.NRBD= 000024       HF.QB = 000004       L$$11R= 000000
CL.RNE= 000004       FE.DRV= 000010       F.NREC= 000030       HF.UBM= 000001       MGEXT   003644R
CL.XIT= 100000       FE.DYM= 010000       F.OVBS= 000030       I$$RAR= 000000       MGMGE   003656R
CMODE = 140000       FE.EXP= 000200       F.RACC= 000016       I$$RDN= 000000       MPAR  = 172100
CRAVL   003310R      FE.EXT= 000001       F.RATT= 000001       KDSAR0= 172360       MPCSR = 177746
CRCOV   004104R      FE.EXV= 000004       F.RCNM= 000034       KDSDR0= 172320       M$$CRB= 000124
CS.DIF= 000002       FE.FDT= 002000       F.RCTL= 000017       KINAR0= 172340       M$$CRX= 000000
CS.DVF= 000004       FE.LSI= 000400       F.RSIZ= 000002       KINAR5= 172352       M$$FCS= 000000
CS.EQU= 000040       FE.MUP= 000002       F.RTYP= 000000       KINAR6= 172354       M$$MGE= 000000
CS.INP= 000001       FE.MXT= 040000       F.SEQN= 000100       KINAR7= 172356       M$$NET= 000000
CS.MOR= 000020       FE.NLG= 100000       F.SPDV= 000072       KISAR0= 172340       M$$OVR= 000000
CS.NMF= 000001       FE.OFF= 001000       F.SPUN= 000074       KISAR5= 172352       NETPF   004022R
CS.OUT= 000002       FE.PKT= 000100       F.STBK= 000036       KISAR6= 172354       NS0     003524R
CS.WLD= 000010       FE.PLA= 000020       F.UNIT= 000136       KISAR7= 172356       NS1     003536R
C$$CKP= 000000       FE.X25= 004000       F.URBD= 000020       KISDR0= 172300       NS2     003550R
C$$ORE= 000400       FILE  = ******  GX   F.VBN = 000064       KISDR6= 172314       NS3     003562R
C$$RSH= 177564       FMASK   003360R      F.VBSZ= 000060       KISDR7= 172316       NS4     003574R
C.CMLD= 000002       FO.MFY= ******  GX   F2.ACN= 000020       K$$CNT= 177546       NS5     003606R
C.DEVD= 000006       FRKHD   003442R      F2.AHR= 010000       K$$CSR= 177546       NS6     003620R
C.DIRD= 000012       FX.DIR= 000004       F2.DAS= 000001       K$$LDC= 000000       NS7     003632R
C.DSDS= 000006       FX.DLN= 000003       F2.DPR= 000400       K$$TPS= 000074       NTUIC   004034R
C.FILD= 000016       FX.DMX= 000002       F2.EVT= 000010       LB$REV= 000400       N$$ACC= 000001
C.MKW1= 000024       FX.LEN= 000000       F2.GGF= 002000       LD$ACC= 100000       N$$BUF= 000001
C.MKW2= 000026       F$$LVL= 000001       F2.IRR= 001000       LD$CLS= 020000       N$$LDV= 000001
```

```
                                                                      F 13
VINI - VNP INITIALIZATION       MACRO V05.03b Saturday 29-Jun-85 02:43  Page 14-2
```

E 14

```
VLBIO - VNP LIBRARY BLOCK I/O R MACRO V05.03b Saturday 29-Jun-85 02:44  Page 7-1
Symbol table

A$$CHK= 000000        F.RSIZ= ****** GX     L$$P11= 000001        N$$PEM= 000001        $BFR  = ****** GX
A$$CPS= 000000        F.URBD= ****** GX     L$$11R= 000000        P$$P45= 000000        $GETAD= ****** GX
A$$PRI= 000000        G$$TPP= 000000        M$$CRB= 000124        P$$WRD= 000000        $GETLB  000000RG
A$$TRP= 000000        G$$TSS= 000000        M$$CRX= 000000        Q$$OPT= 000010        $GETRV= ****** GX
C$$CKP= 000000        G$$TTK= 000000        M$$FCS= 000000        R$$DER= 000000        $NTLHB= ****** GX
C$$ORE= 000400        G$$WRD= 000000        M$$MGE= 000000        R$$K11= 000001        $NTLPT= ****** GX
C$$RSH= 177564        I$$RAR= 000000        M$$NET= 000000        R$$SND= 000000        $PUTAD= ****** GX
D$$BUG= 177514        I$$RDN= 000000        M$$OVR= 000000        R$$11M= 000000        $PUTLB  000116RG
D$$ISK= 000000        K$$CNT= 177546        N$$ACC= 000001        SYSFDB= ****** GX     $PUTRC= ****** GX
D$$L11= 000001        K$$CSR= 177546        N$$BUF= 000001        S$$WRG= 000000        $RADDR= ****** GX
D$$YNC= 000000        K$$LDC= 000000        N$$LDV= 000001        S$$YSZ= 007600        $RBLCK= ****** GX
D$$YNM= 000000        K$$TPS= 000074        N$$MCP= 000001        T$$KMG= 000000        .LBIAS  000000RG     002
E$$XPR= 000000        LD$LP = 000000        N$$MLL= 000001        T$$MIN= 000000        .LBUFF  000002RG     002
F$$LVL= 000001        L$$ASG= 000000        N$$MOV= 000010        V$$CTR= 001000        .MGMGE= ****** GX
F.NRBD= ****** GX     L$$DRV= 000000        N$$NCT= 000001        X$$DBT= 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000220    001   (RW,I,LCL,REL,CON)
DATA    000102    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9493 Words  ( 38 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.15
SY:VLBIO.V2,[132,134]VLBIO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VLBIO
```

F 14

```
VLBIO       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 1
```

```
000706  062716  000002          10$:    ADD     #2,(SP)         ; Else, reject
000712  000407                          BR      30$             ; And return

000714                          20$:    CALL    CKEVE           ; Check event types
000720  103772                          BCS     10$             ; If cs, error
000722  005200                          INC     R0              ; Go to next event type
000724  020067  000000G                 CMP     R0,.PNUMB       ; Is this still in range ?
000730  003771                          BLE     20$             ; If le, yes .. continue
000732                          30$:    RETURN

                                ;
                                ; Check LOGGING entities
                                ;
000734                          $CKLOG::
000734  016700  000000G                 MOV     $OPTON,R0       ; Get <nod,lin,cir> bits of $opton.
000740  042700  173717                  BIC     #^C<OP$NOD!OP$LNE!OP$CIE>,R0

                                ;       R0 := BIC(-R0,R0)       ; Strips the lo-order bit of R0.
000744  010001                            MOV   R0,R1
000746  005401                            NEG   R1
000750  040100                            BIC   R1,R0

000752                          10$:;   IF any bits left
000752  001412                            BEQ   19$
                                ;       THEN fail               ; More than one entity was specified.
000754  062716  000002                    ADD   #2,(SP)
000760                                    ERROR$ BDGRP          ; Bad parameter grouping.
001000                          19$:;   ENDIF

001000                                  RETURN

                                ;
                                ; Object task copies
                                ;
001002                          $COP::
001002  005767  000000G                 TST     .PNUMH          ; Must be a word value
001006  001013                          BNE     101$            ; If ne, illegal
001010  016767  000000G 000010G         MOV     .PNUMB,RQB+LO.COP ; Store the number of copies
001016  022767  000077  000000G         CMP     #MXCOP,.PNUMB   ; Is the value in range ?
001024  103404                          BLO     101$            ; If lo, no
001026  001002                          BNE     10$             ; If ne, not maximum
001030  005067  000010G                 CLR     RQB+LO.COP      ; Else, set to maximum
001034                          10$:    RETURN
001036                          101$:   ERROR$  OCOPTL          ; Illegal number of object copies

                                ;
                                ; Line cost
                                ;
001056  005767  000000G         $COST:: TST     .PNUMH          ; Must be a word value
001062  001011                          BNE     101$            ; If ne, no
001064  026727  000000G 000031          CMP     .PNUMB,#MXCST   ; Check cost value
001072  101005                          BHI     101$            ; If hi, illegal cost value
001074  016767  000000G 000020G         MOV     .PNUMB,RQB+LO.COS ; Else, save cost value
001102  001401                          BEQ     101$            ; If zero, illegal
001104                                  RETURN
001106                          101$:   ERROR$  ILLCOS          ; Illegal cost value

```

```
004770  162602                          SUB     (SP)+,R2        ; Restore output buffer ptr.
004772  016701  000000G                 MOV     .PSTCN,R1       ; Restore node name length.
004776  112022                  20$:    MOVB    (R0)+,(R2)+     ; Save remote node name
005000  005301                          DEC     R1              ;       ...
005002  003375                          BGT     20$             ;       ...

005004                                  RETURN
005006                          101$:   ERROR$  NAMETL          ; Node name too long

                                ;
                                ; Partition name
                                ;
005026  012702  000006          $STPAR::MOV     #6,R2           ; Set max field length
005032                                  CALL    RAD             ; Convert up to 6 characters
005036  103405                          BCS     101$            ; If CS, more than 6
005040  010067  000010G                 MOV     R0,RQB+LO.PAR   ; Set partition name
005044  010167  000012G                 MOV     R1,RQB+LO.PAR+2 ;       ...
005050                                  RETURN
005052                          101$:   ERROR$  PRNMTL

                                ;
                                ; Receive pas. ord
                                ;
005072  012702  000030G         $STRPA::MOV     #RQB+LO.RPA,R2  ; Destination field address
005076  016700  000000G                 MOV     .PSTPT,R0       ; Get field address
005102  016701  000000G                 MOV     .PSTCN,R1       ; And length
005106  003406                          BLE     12$             ; Null field ?
005110  020127  000010                  CMP     R1,#8.          ; Too long ?
005114  003014                          BGT     101$            ; Yes
005116  112022                  10$:    MOVB    (R0)+,(R2)+     ; Move all characters
005120  005301                          DEC     R1              ;       ...
005122  001375                          BNE     10$             ; Loop till done
005124  012701  000010          12$:    MOV     #8.,R1          ; Max length
005130  166701  000000G                 SUB     .PSTCN,R1       ; Number of zeros
005134  003403                          BLE     40$             ; Nothing to pad if zero
005136  105022                  15$:    CLRB    (R2)+           ; Pad with binary zero(s)
005140  005301                          DEC     R1              ;       ...
005142  001375                          BNE     15$             ;       ...
005144                          40$:    RETURN
005146                          101$:   ERROR$  PASWTL          ; Password too long

                                ;
                                ; Sink name
                                ;
005166  016700  000000G         $STSNM::MOV     .PSTCN,R0       ; Get number of bytes in sink name
005172  003403                          BLE     10$             ; Br if invalid count
005174  020027  000050                  CMP     R0,#MXSNKN      ; Valid number of bytes in name?
005200  101403                          BLOS    20$             ; Br if not valid
005202  062716  000002          10$:    ADD     #2,(SP)         ; Reject the transition
005206  000407                          BR      40$             ; And exit
005210  012702  000035G         20$:    MOV     #RQB+LO.SNM,R2  ; Point to storage for sink name
005214  016701  000000G                 MOV     .PSTPT,R1       ; Point to address of sink name
005220  112122                  30$:    MOVB    (R1)+,(R2)+     ; Save sink name
005222  005300                          DEC     R0              ; Decrement count
005224  003375                          BGT     30$             ; Loop until finished
005226                          40$:    RETURN

```

```
                                         .SBTTL  ALLXCB - ALLOCATE X25 CIRCUIT BLOCK

                                 ;+
                                 ;
                                 ; ALLXCB - ALLOCATE AND SET UP X25 CIRCUIT BLOCK
                                 ;
                                 ; INPUTS:
                                 ;          PRTADD - ADDRESS OF ENTRY IN PORT TABLE
                                 ;          $PFLAG - PSI characteristics word                                      ;[TD001]
                                 ;
                                 ; OUTPUTS:
                                 ;          R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK (APR 5 MAPPED TO BLOCK)
                                 ;          R0,R1,R2 DESTROYED
                                 ;
                                 ;-
000440  012701  000106           ALLXCB: MOV     #X$LEN,R1         ; GET SIZE OF X.25 CIRCUIT BLOCK
000444  032767  000000G 000000G          BIT     #PF.PSV,$PFLAG    ; Check for PSV module                         ;[TD001]
000452  001402                           BEQ     5$                ; Branch if not present                        ;[TD001]
000454  012701  000106                   MOV     #X$GLEN,R1        ; PSV needs longer XCB                         ;[TD001]
000460                           5$:     CALL    $ALPOL            ; ALLOCATE CIRCUIT BLOCK                       ;[TD001]
000464  103476                           BCS     50$               ; BR IF UNSUCCESSFUL ALLOCATION                ;**-1
000466  016702  000000G                  MOV     $PSIPT,R2         ; POINT TO PSI HOME BLOCK
000472  062702  000020                   ADD     #H$PTB,R2         ; POINT TO PORT TABLE LISTHEAD
000476                                   GETRV   R2,#2             ; READ IN PORT TABLE LISTHEAD
000514                                   SAVMAP                    ; SAVE CURRENT MAPPING
000520  016701  000000G                  MOV     PORTNO,R1         ; GET NUMBER OF PORTS
000524  006301                           ASL     R1                ; GET NUMBER OF BYTES IN PORT TABLE
000526  016746  000000G                  MOV     $BFR,-(SP)        ; SAVE UNMAPPED PORT TABLE ADDRESS
000532                                   CALL    $CEACC            ; MAP TO PORT TABLE
000536                                   GETAD   (SP)+,R1          ; READ IN PORT TABLE
000552  010077  000000'                  MOV     R0,@PRTADD        ; STORE UNMAPPED ADDRESS IN PORT TABLE
000556                                   PUTAD                     ; WRITE OUT UPDATED PORT TABLE
000562                                   RESMAP                    ; RESTORE PREVIOUS MAPPING
000566                                   CALL    INIXCB            ; INITIALIZE X25 CIRCUIT BLOCK
000572                                   CALL    FNDHSH            ; FIND HASH TABLE ENTRY AND SET UP
000576  103431                           BCS     50$               ; BR IF NO ENTRY AVAILABLE
000600  012700  000106                   MOV     #X$LEN,R0         ; Get length of circuit block                  ;[TD001]
000604  032767  000000G 000000G          BIT     #PF.PSV,$PFLAG    ; Check for PSV module                         ;[TD001]
000612  001402                           BEQ     8$                ; Branch if not present                        ;[TD001]
000614  012700  000106                   MOV     #X$GLEN,R0        ; PSV needs longer XCB                         ;[TD001]
000620                           8$:     GETAD   R3,R0             ; Read in circuit block                        ;[TD001]
000634  016700  000014'                  MOV     PVCOWN,R0         ; GET PROCESS OWNER NAME                       ;**-1
000640  001404                           BEQ     40$               ; IF EQ, NONE SPECIFIED
000642                                   CALL    FNDPDV            ; GET PDV INDEX
000646  110167  000023G                  MOVB    R1,X$MOWN+$BFR    ; STORE PDV INDEX
000652  005067  000030G          40$:    CLR     X$TCLZ+$BFR       ; STORE TIME SINCE LAST ZEROED
000656                                   PUTAD                     ; WRITE OUT UPDATED CIRCUIT BLOCK
000662                           50$:    RETURN
```

```
VFPPVC

SYMBOL

SYMBOL

PF.XDF
PORTNO
PRTADD
PRTBIA
PVCCTM
PVCDF
PVCFLG
PVCLCN
PVCNAM
PVCNML
PVCOWN
PVCPRT
PVCPSZ
PVCST
PVCT
PVCTL
PVCWSZ
PVDLM
PVEND
PVFG
PVLCN
PVNAM
PVNAMX
PVOWN
PVPSZ
PVSTA
PVWSZ
RTSPC
SPACE
SP$P4
STORID
SYNERR
SYSFDB

S$$BAS
WNDSZD
XT$PVC
X$ABQ
X$CTIM
X$DTE
X$GLEN
X$LCN
X$LEN
X$MOWN
X$PKSZ
X$PRT
X$RXQ
X$ST
X$TCLZ
X$TXQ
X$TYP
X$WAQ
```

F 2

```
VFPPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:38        PAGE 2
SYMBOL CROSS REFERENCE                                       CREF    04.00

SYMBOL   VALUE              REFERENCES

PF.XDF   = ******  GX       13-386
PORTNO   = ******  GX       14-437    19-657    19-671
PRTADD     000000 R        #7-116     14-442   *19-668
PRTBIA     000002 R        #7-117    *19-667
PVCCTM     000006 R        #7-119    *11-317    20-709
PVCDF      000000 R         9-169
PVCFLG     000025 R        #7-128    *11-289   *12-371    17-564    18-614
PVCLCN     000004 R        #7-118    *11-305    17-565    18-615    20-712    21-749    21-766
PVCNAM     000027 R        #7-130     15-489    16-525    18-612    23-805
PVCNML     000026 R        #7-129     11-282    15-486    17-552    17-560    17-568
PVCOWN     000014 R        #7-122    *11-288   *12-357    14-453
PVCPRT     000024 R        #7-127    *19-669    20-711    21-767
PVCPSZ     000010 R        #7-120    *11-286   *12-330    20-708
PVCST      000000 RG       #10-190
PVCT       000166 R        #11-315
PVCTL      002344 R        #23-824
PVCWSZ     000012 R        #7-121    *11-287   *12-342    20-707
PVDLM      002276 R        #23-805
PVEND      000352 R         12-372   #13-383
PVFG       000314 R        #12-367
PVLCN      000130 R        #11-299
PVNAM      000056 R        #11-281
PVNAMX   = 000006          #6-100     11-283    16-524    18-611
PVOWN      000256 R        #12-352
PVPSZ      000212 R        #12-328
PVSTA      002354 R        #23-826
PVWSZ      000234 R        #12-340
RTSPC    = ******  GX       8-145     8-146     8-147     8-148
SPACE    = 000040          #6-99
SP$P4      000054 RG       #7-140     20-721
STORID   = ******  GX       11-284
SYNERR   = ******  GX      *9-174    *9-177
SYSFDB   = ******  GX      *14-435   *14-441   *14-452   *15-479   *15-485   *16-517   *16-523   *17-570   *18-618
                           *19-655   *19-659   *20-700   *21-757
S$$BAS   = ******           8-145     8-145     8-146     8-146     8-147     8-147     8-148     8-148
WNDSZD   = ******  GX       11-287
XT$PVC   = 000001           20-722
X$ABQ      000076          *20-719    20-720   *20-720
X$CTIM     000060          *20-709
X$DTE      000024          *20-710
X$GLEN     000106           14-430    14-451    20-699
X$LCN      000026          *20-712
X$LEN      000106           14-427    14-448    20-696
X$MOWN     000023          *14-456
X$PKSZ     000016          *20-708
X$PRT      000021          *20-711
X$RXQ      000072          *20-715    20-716   *20-716
X$ST       000005          *20-721
X$TCLZ     000030          *14-457
X$TXQ      000066          *20-713    20-714   *20-714
X$TYP      000015          *20-722
X$WAQ      000062          *20-717    20-718   *20-718
```

```
VFPRDT -
RDT$DF A
    223
    224
    225
    226
    227
    228
    229
    230
    231
    232
    233
    234
    235
    236
    237
    238
    239
    240
    241
    242
    243
    244
    245
    246
    247
    248
    249
    250
    251
    252
    253
    254
    255
    256
    257
    258
    259
    260
    261
    262
    263
    264
    265
    266
    267
    268
    269
    270
    271
    272
```

G 2

```


                                        ;
                                        ; END OF REMOTE DTE ADDRESS
                                        ;
000212  004567  000000G                 RDEND:  JSR     R5,$SAVRG               ; SAVE R3-R5
000216  012700  000010                          MOV     #<RDTLN+1>/2,R0         ; GET LENGTH OF PACKED ADDRESS
000222  012704  000006'                         MOV     #RDTE,R4                ; GET ADDRESS OF REMOTE DTE ADDR BUFFER
000226  012705  000025'                         MOV     #RDTEP,R5               ; GET ADDRESS OF PACKED ADDRESS BUFFER
000232                                          CALL    PCKBCD                  ; PACK ADDRESS IN BCD FORMAT
000236                                          CALL    CHKRDT                  ; CHECK DTE NAME FOR UNIQUENESS
000242  103474                                  BCS     101$                    ; BR IF NOT UNIQUE
000244  012701  000024                          MOV     #R$LEN,R1               ; GET LENGTH OF BLOCK TO ALLOCATE

000250                                          CALL    $SAVAL                  ; SAVE ALL REGISTERS
000254                                          CALL    $ALPOL                  ; ALLOCATE FROM NETWORK POOL
000260  103470                                  BCS     102$                    ; BR IF UNSUCCESSFUL ALLOCATION
000262                                  10$:    SAVMAP                          ; SAVE CURRENT MAPPING
000266  010005                                  MOV     R0,R5                   ; SAVE UNMAPPED ADDRESS
000270                                          CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
000300  012702  000000G                         MOV     #$BFR,R2                ; GET ADDRESS OF BLOCK
000304  105022                          15$:    CLRB    (R2)+                   ; INITIALIZE REMOTE DTE NAME BLOCK
000306  005301                                  DEC     R1                      ; MORE TO INITIALIZE?
000310  003375                                  BGT     15$                     ; BR IF YES
000312  012701  000000G                         MOV     #$BFR,R1                ; GET ADDRESS OF BLOCK
000316  062701  000002                          ADD     #R$NAM,R1               ; POINT TO REMOTE DTE NAME
000322  012702  000000'                         MOV     #RDTNAM,R2              ; POINT TO SPECIFIED NAME
000326  012703  000006                          MOV     #RDTEMX,R3              ; GET LENGTH OF NAME
000332  112221                          20$:    MOVB    (R2)+,(R1)+             ; STORE REMOTE DTE NAME
000334  005303                                  DEC     R3                      ; MORE TO STORE?
000336  003375                                  BGT     20$                     ; BR IF YES
000340  116767  000000G 000013G                 MOVB    COUNT,R$DAL+$BFR        ; STORE LENGTH OF DTE ADDRESS
000346  012701  000010                          MOV     #<RDTLN+1>/2,R1         ; GET LENGTH OF DTE ADDRESS
000352  012702  000025'                         MOV     #RDTEP,R2               ; GET ADDRESS OF REMOTE DTE ADDRESS
000356  012703  000000G                         MOV     #$BFR,R3                ; POINT TO REMOTE DTE NAME BLOCK
000362  062703  000014                          ADD     #R$DTE,R3               ; POINT TO DTE ADDRESS FIELD
000366  112223                          30$:    MOVB    (R2)+,(R3)+             ; STORE REMOTE DTE ADDRESS
000370  005301                                  DEC     R1                      ; MORE TO STORE?
000372  003375                                  BGT     30$                     ; BR IF YES
000374                                          PUTAD   R0,#R$LEN               ; WRITE UPDATED REMOTE DTE BLOCK
000412  016700  000000G                         MOV     $PSIPT,R0               ; GET ADDRESS OF PSI HOME BLOCK
000416  062700  000004                          ADD     #H$RDTE,R0              ; POINT TO REMOTE DTE LISTHEAD
000422                                          CALL    LNKEND                  ; LINK AT END OF LIST
000426                                          RESMAP                          ; RESTORE PREVIOUS MAPPING
000432                                          RETURN                          ;

                                        ;
                                        ; ERRORS
                                        ;
000434                                  101$:   EMSG$R  L0                      ; ILLEGAL REMOTE DTE NAME
000442                                  102$:   EMSG$R  L2                      ; REMOTE DTE BLOCK ALLOCATION FAILURE
```

```
                                        .SBTTL  $QRNW/$QRNA - Look for RNW$DF/RNA$DF macro

                                ;+
                                ; *** $QRNW/$QRNA - Look for RNW$DF/RNA$DF macro
                                ;
                                ; INPUTS:
                                ;       None.
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=success/failure
                                ;       R3,R4,R5 destroyed
                                ;-
                                        .ENABL  LSB
000000  012705  000000'         $QRNW:: MOV     #RNWDF,R5       ; State table address
000004  000402                          BR      10$

000006  012705  000040'         $QRNA:: MOV     #RNADF,R5       ; State table address

000012                          10$:    VNPDBG  VFPRNW          ; VNP debugging break point
000012  012701  000001                  MOV     #1,R1           ; Full keyword match length, blanks significant
000016  012702  000000'                 MOV     #RNWKW,R2       ; Keyword table address
000022  016703  000000G                 MOV     CFGSZ,R3        ; Record length
000026  012704  000000G                 MOV     #CFGBF,R4       ; Record buffer address
000032  005067  000000G                 CLR     SYNERR          ; Clear syntax error flag
000036                                  CALL    .TPARS          ; Go do the parse
000042  103003                          BCC     20$             ; If CC, found what we were looking for
000044  005367  000000G                 DEC     SYNERR          ; Did syntax error occur?
000050  001401                          BEQ     101$            ; If EQ, yes
000052                          20$:    RETURN
                                ;
                                ; Error condition
                                ;
000054                          101$:   EMSG$R  1T              ; SYNTAX ERROR
                                        .DSABL  LSB
```

```
    492 001004                        100$:   RESMAP                          ; Restore mapping
    493 001010                                RETURN
```

```
                                        .SBTTL  LOCAL DATA STORAGE AND ERROR MESSAGES

  000000                                .PSECT  DATA,D

                                ;
                                ; ERROR MESSAGES
                                ;

                                        .ENABL  LC

  000000                                NTLER$  ,9A,10,CERR,RTSPC,CFLIN,<occurrence of X29$DF>
  000036                                NTLER$  ,9B,10,CERR,RTSPC,CFLIN,<value for max circuits>
  000076                                NTLER$  ,9C,10,CERR,RTSPC,CFLIN,<counter timer value>
  000132                                NTLER$  ,9D,8,CERR,RTSPC,CFLIN,<X.29 data block allocation failure>

                                        .DSABL  LC


                                ;
                                ; LOCAL STORAGE FOR X29$DF PARAMETERS
                                ;
  000206                        MAXVC:  .BLKW   1               ; MAXIMUM CIRCUITS
  000210                        CTIMR:  .BLKW   1               ; COUNTER TIMER VALUE

  000000                                .PSECT
```

VFPX3P
LOCAL

```
     78                                          .SBTTL  LOCAL SYMBOL DEFINITIONS
     79
     80                                  ;
     81                                  ; LOCAL SYMBOL DEFINITIONS
     82                                  ;
     83         000001                           CHNLMN = 1.            ; MINIMUM VALUE FOR CHANNEL RANGE
```

```
**FI

VV
VV
VV
VV
VV
VV
VV
VV
VV
VV
  VV
  VV

LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LLLL
LLLL
```

F 8

```
**FILE**ID**VHDAT

VV      VV  HH      HH  DDDDDDDD      AAAAAA    TTTTTTTTTT
VV      VV  HH      HH  DDDDDDDD      AAAAAA    TTTTTTTTTT
VV      VV  HH      HH  DD      DD  AA      AA      TT
VV      VV  HH      HH  DD      DD  AA      AA      TT
VV      VV  HH      HH  DD      DD  AA      AA      TT
VV      VV  HH      HH  DD      DD  AA      AA      TT
VV      VV  HHHHHHHHHH  DD      DD  AA      AA      TT
VV      VV  HHHHHHHHHH  DD      DD  AA      AA      TT
VV      VV  HH      HH  DD      DD  AAAAAAAAAA      TT
VV      VV  HH      HH  DD      DD  AAAAAAAAAA      TT
  VV  VV    HH      HH  DD      DD  AA      AA      TT            ....
  VV  VV    HH      HH  DD      DD  AA      AA      TT            ....
    VV      HH      HH  DDDDDDDD    AA      AA      TT            ....
    VV      HH      HH  DDDDDDDD    AA      AA      TT            ....


LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
VHELP

SYMBOL

SYMBOL

LD.ALI
LD.CEX
LD.CIR
LD.LIN
LD.LOG
LD.MOD
LD.NOD
LD.OBJ
LD.PRO
LO.ADD
LO.ADR
LO.CLS
LO.CON
LO.COP
LO.COS
LO.CSR
LO.CTL
LO.DES
LO.DTE
LO.EVT
LO.GRO
LO.HOS
LO.HTM
LO.INA
LO.INC
LO.LDT
LO.LID
LO.LIN
LO.LTM
LO.MAC
LO.MDE
LO.NAM
LO.NET
LO.NNM
LO.OFL
LO.ONM
LO.OUT
LO.OWN
LO.PAR
LO.PRI
LO.RET
LO.RPA
LO.SCR
LO.SEC
LO.SKA
LO.SKN
LO.SNM
LO.TPA
LO.TRE
LO.TRI
LO.UNI
LO.VCT
```

G 8

F 9

```
VHELP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:42      PAGE 1
SYMBOL  CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE               REFERENCES

LD.ALI  = 000030            #5-68
LD.CEX  = 000024            #5-68
LD.CIR  = 000035            #5-68
LD.LIN  = 000025            #5-68
LD.LOG  = 000031            #5-68
LD.MOD  = 000037            #5-68
LD.NOD  = 000027            #5-68
LD.OBJ  = 000032            #5-68
LD.PRO  = 000026            #5-68
LO.ADD    000022            #5-68
LO.ADR    000052            #5-68
LO.CLS    000002            #5-68
LO.CON    000006            #5-68
LO.COP    000010            #5-68
LO.COS    000020            #5-68
LO.CSR    000010            #5-68       5-68
LO.CTL    000032            #5-68
LO.DES    000002            #5-68       5-68
LO.DTE    000047            #5-68
LO.EVT    000004            #5-68
LO.GRO    000024            #5-68
LO.HOS    000040            #5-68
LO.HTM    000032            #5-68
LO.INA    000062            #5-68
LO.INC    000060            #5-68
LO.LDT    000046            #5-68
LO.LID    000030            #5-68
LO.LIN    000004            #5-68
LO.LTM    000034            #5-68
LO.MAC    000022            #5-68
LO.MDE    000024            #5-68
LO.NAM    000044            #5-68
LO.NET  = 000002            #5-68
LO.NNM    000022            #5-68
LO.OFL    000012            #5-68
LO.ONM    000004            #5-68
LO.OUT    000056            #5-68
LO.OWN    000026            #5-68
LO.PAR    000010            #5-68
LO.PRI    000012            #5-68
LO.RET    000064            #5-68
LO.RPA    000030            #5-68
LO.SCR    000016            #5-68
LO.SEG    000054            #5-68
LO.SKA    000014            #5-68
LO.SKN    000014            #5-68
LO.SNM    000035            #5-68
LO.TPA    000020            #5-68
LO.TRB    000034            #5-68
LO.TRI    000030            #5-68
LO.UNT    000033            #5-68
LO.VCT    000014            #5-68
```

```
VIMI
Symbo

A$$CH
A$$CP
A$$PR
A$$TR
CMODE
C$$CK
C$$OR
C$$RS
D$$BU
D$$IS
D$$L1
D$$YN
D$$YN
EROUT
ERR1
ERR2
ERR3
E$$XP
FATAL
FE.CA
FE.CE
FE.DR
FE.DY
FE.EX
FE.EX
FE.EX
FE.FD
FE.LS
FE.MU
FE.MX
FE.NL
FE.OF
FE.PK
FE.PL
FE.X
F$$L
F.RSI
F.URE
F2.A
F2.A
F2.D
F2.D
F2.E
F2.G
F2.I
F2.L.
F2.M

. AB

Erro

***

Work
Work
```

G 9

```
                                                                  F 10
VIMI - VNP READ IN CEX IMAGE    MACRO V05.03b Saturday 29-Jun-85 02:42  Page 5-6
Symbol table

A$$CHK= 000000          F2.POL= 000100          H$LBDA  000070          M$$NET= 000000          UISDR0= 177600
A$$CPS= 000000          F2.RAS= 004000          H$LBDN  000072          M$$OVR= 000000          UISDR4= 177610
A$$PRI= 000000          F2.RBN= 020000          H$LDTE  000002          N$$ACC= 000001          UISDR5= 177612
A$$TRP= 000000          F2.SDW= 000040          H$LEN   000042          N$$BUF= 000001          UISDR6= 177614
CMODE = 140000          F2.STP= 100000          H$LOTS  000032          N$$LDV= 000001          UISDR7= 177616
C$$CKP= 000000          F2.SWP= 040000          H$NETW  000024          N$$MCP= 000001          V$$CTR= 001000
C$$ORE= 000400          F2.WND= 000200          H$NML = 000006          N$$MLL= 000001          X$$DBT= 000000
C$$RSH= 177564          F3.AST= 000200          H$NPT   000022          N$$MOV= 000010          Z.LEN = ****** GX
D$$BUG= 177514          F3.CLI= 001000          H$PTB   000020          N$$NCT= 000001          $ADRX = ****** GX
D$$ISK= 000000          F3.CRA= 000001          H$PVC   000006          N$$PEM= 000001          $ADRZ = ****** GX
D$$L11= 000001          F3.EIS= 000004          H$RDTE  000004          PF.PSV= ****** GX       $BFR  = ****** GX
D$$YNC= 000000          F3.NWK= 000002          H$RNW   000042          PF.XDF= ****** GX       $BIGCT= ****** GX
D$$YNM= 000000          F3.PMN= 004000          H$SVC   000036          PF.XGA= ****** GX       $CEAVL= ****** GX
EROUT   002012R         F3.PRO= 000040          H$TRB   000016          PIRQ  = 177772          $CETAB= ****** GX
ERR1    001266R         F3.RLK= 020000          H$XAVL  000100          PMODE = 030000          $CETDF= ****** GX
ERR2    001344R         F3.SHF= 040000          H$XBIA  000074          PR0   = 000000          $CETLN= ****** GX
ERR3    002004R         F3.STM= 000010          H$X29C  000040          PR1   = 000040          $CETND= ****** GX
E$$XPR= 000000          F3.TCM= 002000          I$$RAR= 000000          PR4   = 000200          $CETST= ****** GX
FATAL = ****** GX       F3.UDS= 000020          I$$RDN= 000000          PR5   = 000240          $CEXND= ****** GX
FE.CAL= 000040          F3.WAT= 010000          KDSAR0= 172360          PR6   = 000300          $CLBUF= ****** GX
FE.CEX= 020000          F3.XHR= 000100          KDSDR0= 172320          PR7   = 000340          $EROUT= ****** GX
FE.DRV= 000010          F3.11S= 000400          KINAR0= 172340          PS    = 177776          $FMASK= ****** GX
FE.DYM= 010000          GCLBUF= ****** GX       KINAR5= 172352          P$$P45= 000000          $GETAD= ****** GX
FE.EXP= 000200          G$$TPP= 000000          KINAR6= 172354          P$$WRD= 000000          $GETRV= ****** GX
FE.EXT= 000001          G$$TSS= 000000          KINAR7= 172356          Q$$OPT= 000010          $LLCTA= ****** GX
FE.EXV= 000004          G$$TTK= 000000          KISAR0= 172340          R$$DER= 000000          $NTLHB= ****** GX
FE.FDT= 002000          G$$WRD= 000000          KISAR5= 172352          R$$K11= 000001          $NTLPT= ****** GX
FE.LSI= 000400          HF$DLM= 000002          KISAR6= 172354          R$$SND= 000000          $PDVNM= ****** GX
FE.MUP= 000002          HF$GWY= 000010          KISAR7= 172356          R$$11M= 000000          $PDVTA= ****** GX
FE.MXT= 040000          HF$HOS= 000004          KISDR0= 172300          SISDR0= 172200          $PFLAG= ****** GX
FE.NLG= 100000          HF$XDF= 000020          KISDR6= 172314          SR0   = 177572          $PSIPT= ****** GX
FE.OFF= 001000          HF.CIS= 000200          KISDR7= 172316          SR3   = 172516          $RADDR= ****** GX
FE.PKT= 000100          HF.EIS= 000002          K$$CNT= 177546          SWR   = 177570          $RBLCK= ****** GX
FE.PLA= 000020          HF.FPP= 100000          K$$CSR= 177546          SYSFDB= ****** GX       $RSIZE= ****** GX
FE.X25= 004000          HF.QB = 000004          K$$LDC= 000000          S$$WRG= 000000          $SAVAL= ****** GX
F$$LVL= 000001          HF.UBM= 000001          K$$TPS= 000074          S$$YSZ= 007600          $SLTMA= ****** GX
F.RSIZ= ****** GX       H$CUG   000010          LD$LF = 000000          TPS   = 177564          $SLTNM= ****** GX
F.URBD= ****** GX       H$DST   000012          L$$ARG= 000000          T$$KMG= 000000          $UNBSZ= ****** GX
F2.ACN= 000020          H$D29   000014          L$$DRV= 000000          T$$MIN= 000000          $VIMI   000000RG
F2.AHR= 010000          H$FLG   000000          L$$P11= 000001          UBMPR = 170200          $$    = 177777
F2.DAS= 000001          H$GLEN  000104          L$$11R= 000000          UDSAR0= 177660          $$$   = 001346R
F2.DPR= 000400          H$GLT   000044          MPAR  = 172100          UDSDR0= 177620          $$$R  = 000010
F2.EVT= 000010          H$GNAM  000050          MPCSR = 177746          UISAR0= 177640          $$$$  = 002004R
F2.GGF= 002000          H$GNML= 000020          M$$CRB= 000124          UISAR4= 177650          .CEAVL= ****** GX
F2.IRR= 001000          H$GPT   000046          M$$CRX= 000000          UISAR5= 177652          .MGMGE= ****** GX
F2.LIB= 000002          H$HITS  000034          M$$FCS= 000000          UISAR6= 177654          .PDVTA= ****** GX
F2.MP = 000004          H$HLEN  000044          M$$MGE= 000000          UISAR7= 177656

. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        002040    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
```

F 11

```
**FILE**ID**VINI



VV      VV     IIIIII     NN      NN      IIIIII
VV      VV     IIIIII     NN      NN      IIIIII
VV      VV       II       NN      NN        II
VV      VV       II       NN      NN        II
VV      VV       II       NNNN    NN        II
VV      VV       II       NNNN    NN        II
VV      VV       II       NN  NN  NN        II
VV      VV       II       NN  NN  NN        II
VV      VV       II       NN    NNNN        II
VV      VV       II       NN    NNNN        II
  VV  VV         II       NN      NN        II           ....
  VV  VV         II       NN      NN        II           ....
    VV         IIIIII     NN      NN      IIIIII         ....
    VV         IIIIII     NN      NN      IIIIII         ....



LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

G 11

```
                                        ;+
                                        ;       **-$SYMOV
                                        ;
                                        ;       MOVE DEFINED SYMBOLS TO ROOT SECTION ( VRES ) REPORTING
                                        ;       FATAL ERRORS TO USER.
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ; OUTPUTS:
                                        ;       R0,R1,R2 DESTROYED
                                        ;       C-BIT SET IF FATAL UNDEFINED REFERENCE ENCOUNTERED
                                        ;-

002264  006146                          SYMOV:  ROL     -(SP)           ; C-BIT MUST BE CLEAR ON ENTRY
002266  012700  000000G                         MOV     #$SYMTB,R0      ; DESTINATION TABLE
002272  012701  003276'                         MOV     #EXSTB,R1       ; SOURCE TABLE

002276  005761  000002                  5$:     TST     SB.DEF(R1)      ; SYMBOL DEFINED ?
002302  001016                                  BNE     20$             ; YES

002304  012702  002632'                         MOV     #OKLIST-2,R2    ; LIST OF SYMBOLS THAT MAY BE UNDEFINED
002310  005722                          10$:    TST     (R2)+           ; INCREMENT ADDRESS
002312  005712                          11$:    TST     (R2)            ; END OF LIST ?
002314  001426                                  BEQ     110$            ; THEN TELL USER OF FATAL ERROR
002316  022261  000004                          CMP     (R2)+,SB.SYB(R1); FIRST HALF MATCH ?
002322  001372                                  BNE     10$             ; NO
002324  022261  000006                          CMP     (R2)+,SB.SYB+2(R1) ; SECOND HALF MATCH ?
002330  001370                                  BNE     11$             ; NO
002332  012761  177777  000010          12$:    MOV     #-1,SB.VAL(R1)  ; FLAG SYMBOL AS UNDEFINED

002340  016120  000010                  20$:    MOV     SB.VAL(R1),(R0)+; MOVE SYMBOL TO RESTING PLACE
002344  016101  000000                          MOV     SB.LNK(R1),R1   ; CHAIN DOWN LIST
002350  001352                                  BNE     5$              ; LOOP
                                        ;
                                        ; NOW MOVE COMEXEC SYMBOLS
                                        ;
002352  020027  000000G                         CMP     R0,#$SYMND      ; AT END OF ROOT TABLE YET ?
002356  001403                                  BEQ     30$             ; YES
002360  012701  004046'                         MOV     #CEXSTB,R1      ; LOAD ADDRESS OF CEX SYMBOLS
002364  000744                                  BR      5$              ; MOVE THEM

002366  006026                          30$:    ROR     (SP)+           ; SET/CLEAR C-BIT
002370                                          RETURN
                                        ;
                                        ; ERROR HANDLING
                                        ;
002372  016167  000004  000000G         110$:   MOV     SB.SYB(R1),$BFR         ; LOAD UNDEFINED SYMBOL
002400  016167  000006  000002G                 MOV     SB.SYB+2(R1),$BFR+2     ;  RAD50 NAME
002406  020027  000000G                         CMP     R0,#$SYMND              ; COMEXEC SYMBOL ?
002412  001406                                  BEQ     120$                    ; YES
002414  012746  002444'                         MOV     #130$,-(SP)             ; PUSH RETURN ADDRESS
002420                                          ERRPT$  ERR6,$EROUT             ; EXEC SYMBOL WARNING

002430  012746  002444'                 120$:   MOV     #130$,-(SP)             ; SET RETURN ADDRESS
002434                                          ERRPT$  ERR7,$EROUT             ; CEX SYMBOL WARNING

002444  052716  000001                  130$:   BIS     #1,(SP)                 ; SET C-BIT FLAG
```

```
Symbol table

N$$MCP= 000001      PS.SYS= 000040      SYMOV   002264R     UISAR4= 177650      $PUTRC= ****** GX
N$$MLL= 000001      P$$P45= 000000      SYSFDB= ****** GX   UISAR5= 177652      $RADDR= ****** GX
N$$MOV= 000010      P$$WRD= 000000      SYSIZ   003404R     UISAR6= 177654      $RSIZ = ****** GX
N$$NCT= 000001      P.BLKS  000016      SYSNM   003416R     UISAR7= 177656      $RSIZE= ****** GX
N$$PEM= 000001      P.BUSY  000024      SYSPMT  001314R     UISDR0= 177600      $RSX  = ****** GX
N.ANM1= 000002      P.IOC   000003      S$$WRG= 000000      UISDR4= 177610      $SAVAL= ****** GX
N.ANM2= 000024      P.LNK   000000      S$$YSZ= 007600      UISDR5= 177612      $STB  = ****** GX
N.DID = 000024      P.MAIN  000012      S.ANM1= 000014      UISDR6= 177614      $SYMBL= ****** GX
N.DVNM= 000032      P.NAM   000004      S.ANM2= 000005      UISDR7= 177616      $SYMND= ****** GX
N.FID = 000000      P.OWN   000026      S.FATT= 000016      UMRHD   003670R     $SYMTB= ****** GX
N.FNAM= 000006      P.PRI   000002      S.FDB = 000140      VCHECK  004130R     $SYSIZ= ****** GX
N.FTYP= 000014      P.REL   000014      S.FNAM= 000006      VCTR    003512R     $VCTOF= ****** GX
N.FVER= 000016      P.SIZE  000016      S.FNB = 000036      VFILE   000202R     $VCTVB= ****** GX
N.NEXT= 000022      P.STAT  000030      S.FNBW= 000017      VF.INI= ****** GX   $VFLAG= ****** GX
N.STAT= 000020      P.SUB   000010      S.FNTY= 000004      VF.SS = ****** GX   $VFLG1= ****** GX
N.UNIT= 000034      P.SWSZ  000022      S.FTYP= 000002      VSYMB   001340R     $VGCL = ****** GX
OKLIST  002634R     P.TCB   000026      S.MPR   003276R     VSYM1   002662R     $VIMI = ****** GX
PARHD   003372R     P.WAIT  000020      S.NFEN= 000020      V$$CTR= 001000      $VINI   000006RG
PATT    003764R     Q$$OPT= 000010      TLGTH = 003740R     V1.NSY= ****** GX   $XBUF2= ****** GX
PDVTA   004046R     R$LDAT  000026      TPS   = 177564      X$$DBT= 000000      $XFRAD= ****** GX
PHDR    003776R     R$LFLG  000024      TSKHD   003714R     $AZFS = ****** GX   $XLEN2= ****** GX
PIRQ  = 177772      R$LHGV  000006      TS$ACP= 020000      $BFR  = ****** GX   $XXBUF= ****** GX
PLGTH   003500R     R$LLDZ  000012      TS$CHK= 000100      $CBOMG= ****** GX   $XXINI= ****** GX
PMODE = 030000      R$LMXV  000010      TS$CMP= 000200      $CEX  = ****** GX   $XXLEN= ****** GX
PPRO    004010R     R$LMXZ  000014      TS$IOP= 000020      $CLBUF= ****** GX   $$    = 177777
PR0   = 000000      R$LNAM  000000      TS$NEW= 000001      $CLCSI= ****** GX   $$$   = 004604R
PR1   = 000040      R$LOFF  000016      TS$NHD= 040000      $CLEVL= ****** GX   $$$$  = 004754R
PR4   = 000200      R$LSA   000004      TS$NSD= 002000      $CLFDB= ****** GX   .C00    003322R
PR5   = 000240      R$LSEG  000022      TS$NXH= 000002      $CLFLG= ****** GX   .CRAVL= ****** GX
PR6   = 000300      R$LSIZ  000034      TS$PIC= 100000      $CLIOS= ****** GX   .CRCOV= ****** GX
PR7   = 000340      R$LWND  000020      TS$PMD= 010000      $CLPMT= ****** GX   .CSI1 = ****** G
PS    = 177776      R$$DER= 000000      TS$PRV= 000400      $CRAVL= ****** GX   .CSI2 = ****** G
PS.APR= 000007      R$$K11= 000001      TS$RES= 000040      $DEVHD= ****** GX   .DEVHD= ****** GX
PS.CHK= 010000      R$$SND= 000000      TS$SLV= 004000      $DFUIC= ****** GX   .DLCOV= ****** GX
PS.CKP= 040000      R$$11M= 000000      TS$SUP= 000010      $EROUT= ****** GX   .EXSIZ= ****** GX
PS.CKR= 020000      R.FIX = ****** GX   TS$XHR= 000004      $EXSIZ= ****** GX   .FMASK= ****** GX
PS.COM= 000200      SB.DEF  000002      T$$KMG= 000000      $FMASK= ****** GX   .NTUIC= ****** GX
PS.DEL= 000010      SB.LNK  000000      T$$MIN= 000000      $GENND= ****** GX   .OPEN = ****** G
PS.DRV= 000020      SB.SYB  000004      T2$CLI= 000001      $GENST= ****** GX   .PARHD= ****** GX
PS.FXD= 004000      SB.VAL  000010      T2$FMP= 000002      $GETRC= ****** GX   .PLGTH= ****** GX
PS.LIO= 001000      SISDR0= 172200      UBMPR = 170200      $GETRV= ****** GX   .PPASC= ****** GX
PS.NSF= 000400      SR0   = 177572      UDSAR0= 177660      $LBXL = 000340      .RDFUI= ****** GX
PS.OUT= 100000      SR3   = 172516      UDSDR0= 177620      $LOWSZ= ****** GX   .SYSIZ= ****** GX
PS.PER= 002000      STKDP   003466R     UICCFG= 000001      $NTUIC= ****** GX   ...TPC= 000006
PS.PIC= 000100      SWR   = 177570      UISAR0= 177640      $PARHD= ****** GX

. ABS.  001000    000    (RW,I,GBL,ABS,OVR)
        005010    001    (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 14098 Words  ( 56 Pages)
Size of core pool: 15496 Words  ( 59 Pages)
```

```
                                                                F 14
VLBIO      CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE             REFERENCES

F.NRBD  = ******   GX     *7-118     *7-124
F.RSIZ  = ******   GX     *6-100     *6-102     *7-121     *7-123
F.URBD  = ******   GX     *6-97      *6-103
SYSFDB  = ******   GX     *6-97      *6-100     *6-102     *6-103     *7-118     *7-121     *7-123     *7-124
$BFR    = ******   GX      6-103      7-124
$GETAD  = ******   GX      6-100
$GETLB    000000 RG       #6-89
$GETRV  = ******   GX      6-102
$NTLHB  = ******   GX      6-90
$NTLPT  = ******   GX      6-91       7-116
$PUTAD  = ******   GX      7-121
$PUTLB    000116 RG       #7-115
$PUTRC  = ******   GX      7-123
$RADDR  = ******   GX     *6-100     *6-102     *7-121     *7-123
$RBLCK  = ******   GX     *6-100     *7-121
.LBIAS    000000 RG       #5-71       6-94      *6-96       7-121      7-123
.LBUFF    000002 RG       #5-76       6-93       7-118
.MGMGE  = ******   GX      6-98       7-119
```

```
                                                                G 14
VLBIO      CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 2
```

```
                                        ;
                                        ; Csr value
                                        ;
001126  005767  000000G                 $CSR::  TST     .PNUMH            ; Must be single precision
001132  001014                                  BNE     101$              ;       ...
001134  022767  160000  000000G                 CMP     #160000,.PNUMB    ; An i/o page address
001142  101010                                  BHI     101$              ;       ...
001144  032767  000003  000000G                 BIT     #3,.PNUMB         ; And a multiple of 4
001152  001004                                  BNE     101$              ;       ...
001154  016767  000000G 000010G                 MOV     .PNUMB,RQB+LO.CSR ; Set csr value
001162                                          RETURN
001164                                  101$:   ERROR$  ILLCSR


                                        ;
                                        ; Controller number
                                        ;
001204  105767  000001G                 $CTL::  TSTB    $QUAL+1           ; Any wild cards ?
001210  001402                                  BEQ     5$
001212  000167  004436                           JMP    ERRLID            ; If ne, yes .. illegal line id
001216  005767  000000G                 5$:     TST     .PNUMH            ; Number must be a byte value
001222  001016                                  BNE     101$              ;       ...
001224  105767  000001G                         TSTB    .PNUMB+1          ;       ...
001230  001013                                  BNE     101$              ;       ...
001232                                          CALL    CHKLOG            ; Check for logging function
001236  103004                                  BCC     10$               ; Br if logging function
001240  116767  000000G 000004G                 MOVB    .PNUMB,RQB+LR.CTL ; Set controller number
001246  000403                                  BR      20$               ; And exit
001250  116767  000000G 000032G         10$:    MOVB    .PNUMB,RQB+LO.CTL ; Set controller number for logging
001256                                  20$:    RETURN
001260                                  101$:   ERROR$  CNUMTL

                                        ;
                                        ; Destination node account field
                                        ;
001300                                  $DACC::
001300  016700  000000G                         MOV     .PSTCN,R0         ; Get account field length
001304  022700  000020                          CMP     #16.,R0           ; Is account field too big ?
001310  103412                                  BLO     101$              ; If lo, yes
001312  016701  000000G                         MOV     .PSTPT,R1         ; Get start address of account field
001316  012702  000052G                         MOV     #RQB+LR.ACO,R2    ; Get address of account buffer
001322  110022                                  MOVB    R0,(R2)+          ; Store field length
001324  001403                                  BEQ     20$               ; If eq, no field specified
001326  112122                          10$:    MOVB    (R1)+,(R2)+       ; Else, store account field
001330  005300                                  DEC     R0                ;       ...
001332  003375                                  BGT     10$               ;       ...
001334                                  20$:    RETURN
001336                                  101$:   ERROR$  ACCTTL            ; Account field too large

                                        ;
                                        ; Destination node password field
                                        ;
001356                                  $DPAS::
001356  016700  000000G                         MOV     .PSTCN,R0         ; Get password field length
001362  022700  000010                          CMP     #8.,R0            ; Is password field too big ?
001366  103412                                  BLO     101$              ; If lo, yes
001370  016701  000000G                         MOV     .PSTPT,R1         ; Get start address of password field
001374  012702  000041G                         MOV     #RQB+LR.PAS,R2    ; Get address of password buffer
```

```
                                        ;
                                        ; Transmit password
                                        ;
005230  012702  000020G                 $STTPA::MOV     #RQB+LO.TPA,R2  ; Destination field address
005234  016700  000000G                         MOV     .PSTPT,R0       ; Get field address
005240  016701  000000G                         MOV     .PSTCN,R1       ; And length
005244  003406                                  BLE     12$             ; Null field ?
005246  020127  000010                          CMP     R1,#8.          ; Too long ?
005252  003014                                  BGT     101$            ; Yes
005254  112022                          10$:    MOVB    (R0)+,(R2)+     ; Move all characters
005256  005301                                  DEC     R1              ;       ...
005260  001375                                  BNE     10$             ; Loop till done
005262  012701  000010                  12$:    MOV     #8.,R1          ; Max length
005266  166701  000000G                         SUB     .PSTCN,R1       ; Number of zeros
005272  003403                                  BLE     40$             ; Nothing to pad if zero
005274  105022                          15$:    CLRB    (R2)+           ; Pad with binary zero(s)
005276  005301                                  DEC     R1              ;       ...
005300  001375                                  BNE     15$             ;       ...
005302                                  40$:    RETURN
005304                                  101$:   ERROR$  PASWTL          ; Password too long

                                        ;
                                        ; Tributary number
                                        ;
                                                .ENABL  LSB
005324  105767  000001G                 $TRI::  TSTB    $QUAL+1         ; Any wild cards specified ?
005330  001151                                  BNE     ERRLID          ; If ne, yes .. illegal line id
005332  005767  000000G                         TST     .PNUMH          ; Number must be a byte value
005336  001027                                  BNE     101$            ;       ...
005340  105767  000001G                         TSTB    .PNUMB+1        ;       ...
005344  001024                                  BNE     101$            ;       ...
005346                                          CALL    CHKLOG          ; Check for logging function
005352  103004                                  BCC     10$             ; Br if logging function
005354  116767  000000G 000006G                 MOVB    .PNUMB,RQB+LR.TRI ; Set tributary number
005362  000403                                  BR      $XTRI           ; And continue
005364  116767  000000G 000030G 10$:    MOVB    .PNUMB,RQB+LO.TRI ; Set tributary number for logging
005372  004267  000000G                 $XTRI:: JSR     R2,$SAVVR       ; Save R0-R2
005376                                          CALL    GETLIN          ; Get SLT and PDV address of line
005402  103415                                  BCS     102$            ; Br if not in system
005404  032762  000000G 000000G                 BIT     #LF.MTP,L.FLG(R2) ; Multipoint device?
005412  001520                                  BEQ     ERRLID          ; Br if no
005414                                          RETURN
005416                                  101$:   ERROR$  TRIBTL
005436                                  102$:   ERROR$  NSLINE          ; Line not in system
                                                .DSABL  LSB


                                        ;
                                        ; Tributary address
                                        ;
005456  005767  000000G                 $TRIBA::TST     .PNUMH          ; Must be a word value
005462  001007                                  BNE     101$            ; If ne, no .. illegal value
005464  105767  000001G                         TSTB    .PNUMB+1        ; Must be a byte value
005470  001004                                  BNE     101$            ; If ne, no .. illeage value
005472  016767  000000G 000030G                 MOV     .PNUMB,RQB+LO.TRI ; Store tributary address
005500                                          RETURN
005502                                  101$:   ERROR$  ILLTRI          ; Illegal tributary address

```

```
                                  .SBTTL  CKPVCO - Check for unique PVC name [new format]          ;[TD001]
                          ;+                                                                         ;[TD001]
                          ;                                                                          ;[TD001]
                          ; *** CKPVCO - Check for unique PVC name [new format]                      ;[TD001]
                          ;                                                                          ;**-5
                          ; INPUTS:
                          ;       NONE
                          ;
                          ; OUTPUTS:
                          ;       CARRY CLEAR - SPECIFIED PVC IS UNIQUE
                          ;       CARRY SET - SPECIFIED PVC IS NOT UNIQUE
                          ;       All registers preserved                                            ;[TD001]
                          ;                                                                          ;[TD001]
                          ;-                                                                         ;[TD001]
000664                    CKPVCO:                                                                    ;[TD001]
000664                            CALL    $SAVAL          ; Save all registers                       ;[TD001]
000670  016700  000000G           MOV     $PSIPT,R0       ; Point to PSI home block                  ;[TD001]
000674  062700  000006            ADD     #H$PVC,R0       ; Point to PVC name block listhead         ;[TD001]
000700                            GETRV   R0,#2           ; Read in listhead                         ;[TD001]
000716                            SAVMAP                  ; Save current mapping                     ;[TD001]
000722  016700  000000G   10$:    MOV     $BFR,R0         ; Get address of next block in list        ;[TD001]
000726  000241                    CLC                     ; Assume success                           ;[TD001]
000730  001431                    BEQ     30$             ; Branch if end of list                    ;[TD001]
000732                            CEACC$  R0              ; Convert to mapped address                ;[TD001]
000742                            GETAD   R0,#C$LEN+MAXID ; Read in maximum-size PVC name block      ;[TD001]
000760  116701  000026'           MOVB    PVCNML,R1       ; Get length of PVC sought                 ;[TD001]
000764  120167  000002G           CMPB    R1,$BFR+C$NML   ; Compare with length of name just read    ;[TD001]
000770  001354                    BNE     10$             ; If not same length, not same PVC         ;[TD001]
000772  012700  000027'           MOV     #PVCNAM,R0      ; Address of PVC sought                    ;[TD001]
000776  012702  000016G           MOV     #$BFR+C$GNAM,R2 ; Address of candidate                     ;[TD001]
001002  122022            20$:    CMPB    (R0)+,(R2)+     ; Check if names match                     ;[TD001]
001004  001346                    BNE     10$             ; Branch if no match                       ;[TD001]
001006  005301                    DEC     R1                                                         ;[TD001]
001010  003374                    BGT     20$                                                        ;[TD001]
001012  000261                    SEC                     ; Else error - set c-bit                   ;[TD001]
001014                    30$:    RESMAP                  ; Restore previous mapping                 ;[TD001]
001020                            RETURN                                                             ;[TD001]
```

```
VFPPVC

SYMBOL

SYMBOL

X$WSZ
$ALPHA
$ALPOL
$ANY
$BFR



$BLANK
$CAT5
$CEACC
$DBG$
$DIGIT
$DNUMB
$EOS
$ERRN0
$ERRN1
$ERRN2
$ERRN3
$ERRO0
$ERRO5
$ERRO9
$ERRP8
$ERR1T
$EXIT
$FAIL
$GETAD
$GETRV
$GPRM
$LAMDA
$NUMBR
$PFLAG
$PSIPT
$PUTAD
$QPVC
$RADDR

$RAD50
$RBLCK

$RONLY
$SAVAL
$STRNG
$SUBXP
$$$FLG
$$$KEY
.PNUMB
.PNUMH
.PSTCN
.PSTPT
.TPARS
```

```
VFPPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:38      PAGE 3
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE               REFERENCES

X$WSZ     000020            *20-707
$ALPHA  = 000022            #10-190
$ALPOL  = ******   GX        14-431    17-554    18-605
$ANY    = 000020            #10-190
$BFR    = ******   GX        14-439   *14-456   *14-457    15-481    15-487    15-490    16-519    16-526    17-559
                            *17-562   *17-564   *17-565   *17-566   *17-567    18-610   *18-614   *18-615   *18-616
                            *18-617    19-656    19-660    20-701   *20-707   *20-708   *20-709   *20-710   *20-711
                            *20-712   *20-713   *20-714   *20-715   *20-716   *20-717   *20-718   *20-719   *20-720
                            *20-721   *20-722    21-758   *21-766   *21-767
$BLANK  = 000006            #10-190
$CAT5   = ******   GX        12-356    23-806
$CEACC  = ******   GX        14-440    15-484    16-522    17-558    18-609    19-656    20-695    21-746
$DBG$   = ******              9-167
$DIGIT  = 000024            #10-190
$DNUMB  = 000014            #10-190
$EOS    = 000012            #10-190
$ERRNO    000056 R          #8-145     11-294    13-409    23-819
$ERRN1    000116 R          #8-146     12-362
$ERRN2    000152 R          #8-147     12-377
$ERRN3    000204 R          #8-148     13-410
$ERR00  = ******   GX        12-347
$ERR05  = ******   GX        11-322
$ERR09  = ******   GX        11-310
$ERRP8  = ******   GX        12-335
$ERR1T  = ******   GX         9-184
$EXIT   = 000000            #10-190
$FAIL   = 177777            #10-190
$GETAD  = ******   GX        14-441    14-452    15-485    16-523    19-659    20-700    21-757
$GETRV  = ******   GX        14-435    15-479    16-517    19-655
$GPRM   = ******             10-190
$LAMDA  = 000000            #10-190
$NUMBR  = 000002            #10-190
$PFLAG  = ******   GX        13-386    14-428    14-449    20-697
$PSIPT  = ******   GX        14-433    15-477    16-515    17-579    18-627    19-653
$PUTAD  = ******   GX        14-443    14-458    17-570    18-618    20-723    21-768
$QPVC     000000 RG         #9-167
$RADDR  = ******   GX       *14-435   *14-441   *14-452   *15-479   *15-485   *16-517   *16-523   *17-570   *18-618
                            *19-655   *19-659   *20-700   *21-757
$RAD50  = 000016            #10-190
$RBLCK  = ******   GX        14-436   *14-444    15-480   *15-496    16-518   *16-532    17-550   *17-584    18-602
                            *18-632    19-652    19-667   *19-674    21-744   *21-769
$RONLY  = ******             10-190    10-190    10-190
$SAVAL  = ******   GX        15-476    16-514    17-549    18-601
$STRNG  = 000004            #10-190
$SUBXP  = 000010            #10-190
$$$FLG  = 177777            #10-190
$$$KEY  = 177777            #10-190
.PNUMB  = ******   GX        11-301    11-317    12-369    12-371    23-824    23-826    23-829
.PNUMH  = ******   GX        11-299    11-315    12-367    23-827
.PSTCN  = ******   GX        12-352
.PSTPT  = ******   GX        12-354
.TPARS  = ******   GX         9-175
```

```
VFPRDT -
CHKRDT -
```

```
                                        .SBTTL  CHKRDT - CHECK FOR UNIQUE RDT

                                ;+
                                ;
                                ; CHKRDT - CHECK FOR UNIQUE REMOTE DTE NAME
                                ;
                                ; INPUTS:
                                ;       RDTNAM - SPECIFIED RDT NAME
                                ;
                                ; OUTPUTS:
                                ;       CARRY CLEAR - REMOTE DTE NAME IS UNIQUE
                                ;       CARRY SET - REMOTE DTE NAME IS NOT UNIQUE
                                ;       R0 DESTROYED
                                ;
                                ;-
 000450  016700  000000G        CHKRDT: MOV     $PSIPT,R0               ; GET ADDRESS OF PSI HOME BLOCK
 000454  062700  000004                 ADD     #H$RDTE,R0              ; POINT TO REMOTE DTE LISTHEAD
 000460                                 GETRV   R0,#2                   ; READ IN LISTHEAD
 000476                                 CALL    $SAVAL                  ; SAVE ALL REGISTERS
 000502                                 SAVMAP                          ; SAVE CURRENT MAPPING
 000506  016700  000000G        10$:    MOV     $BFR,R0                 ; GET NEXT BLOCK IN LIST
 000512  000241                         CLC                             ; ASSUME SUCCESS
 000514  001430                         BEQ     25$                     ; BR IF END OF LIST
 000516                                 CEACC$  R0                      ; CONVERT TO MAPPED ADDRES
 000526                                 GETAD   R0,#R$LEN               ; READ IN REMOTE DTE BLOCK
 000544  012701  000000G                MOV     #$BFR,R1                ; GET ADDRESS OF REMOTE DTE BLOCK
 000550  062701  000002                 ADD     #R$NAM,R1               ; POINT TO REMOTE DTE NAME
 000554  012702  000000'                MOV     #RDTNAM,R2              ; POINT TO SPECIFIED REMOTE DTE NAME
 000560  012703  000006                 MOV     #RDTEMX,R3              ; GET LENGTH OF NAME
 000564  122122                 20$:    CMPB    (R1)+,(R2)+             ; IS THE NAME UNIQUE?
 000566  001347                         BNE     10$                     ; BR IF YES
 000570  005303                         DEC     R3                      ; MORE TO CHECK?
 000572  003374                         BGT     20$                     ; BR IF YES
 000574  000261                         SEC                             ; ELSE NOT UNIQUE - SET C-BIT
 000576                         25$:    RESMAP                          ; RESTORE PREVIOUS MAPPING
 000602                         30$:    RETURN                          ;

         000001                         .END
```

```
                      .SBTTL  Parse table for RNW$DF

000062                ISTAT$  RNWST,RNWKW

000062                STATE$  RNWDF
000062                  TRAN$   !SKIPSP                ; Skip leading spaces
000062                STATE$
000062                  TRAN$   %RNW$DF%,,,1,SYNERR    ; Flag RNW$DF recognised

000062                STATE$
000062                  TRAN$   !SKIPSP                ; Skip leading spaces
000062                STATE$
000062                  TRAN$   $RAD50,,RNWNET         ; Parse network name

000062                STATE$
000062                  TRAN$   !OPSEP                 ; Skip operand separator

000062                STATE$
000062                  TRAN$   $RAD50,,RNWNOD         ; Parse node name

000062                STATE$
000062                  TRAN$   !END,$EXIT             ; Ensure correct termination

000062                STATE$
```

```
   495                                            .SBTTL  Parser Subexpression Action Routines
   496                                    ;+
   497                                    ; *** CLRSTR - Clear string length field
   498                                    ;
   499                                    ; INPUTS
   500                                    ;       None
   501                                    ;
   502                                    ; OUTPUTS
   503                                    ;       STRLEN cleared
   504                                    ;-
   505 001012  105067  000164'            CLRSTR: CLRB    STRLEN
   506 001016                                     RETURN
```

```
                                        .SBTTL  LOOK FOR X29$DF MACRO

                                ;+
                                ; $QX29 - LOOK FOR X29$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3,R4,R5=DESTROYED
                                ;-

000000  012705  000000'         $QX29:: MOV     #X29DF,R5       ; STATE TABLE ADDRESS
000004  005001                          CLR     R1              ; FULL KEYWORD MATCH LENGTH
000006  012702  000000'                 MOV     #X29KW,R2       ; KEYWORD TABLE ADDRESS
000012  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
000016  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000022  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000026                                  CALL    .TPARS          ; GO DO THE PARSE
000032  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000034  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000040  001401                          BEQ     101$            ; IF EQ, YES
000042                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000044                          101$:   EMSG$R  1T              ; SYNTAX ERROR
```

```
                                      .SBTTL  LOCAL DATA

        000000                        .PSECT  DATA,D

                              ;
                              ; LOCAL DATA FOR CHN$DF MACRO
                              ;
        000000                CHNLLO: .BLKW   1               ; LOW END OF CHANNEL RANGE
        000002                CHNLHI: .BLKW   1               ; HIGH END OF CHANNEL RANGE

                              ;
                              ; ERROR MESSAGES
                              ;
                                      .ENABL  LC

        000004                        NTLER$  ,08,8,CERR,RTSPC,CFLIN,<Channel block allocation failure>

                                      .DSABL  LC

        000000                        .PSECT


```

```
                                .TITLE  VHDAT - HELP SUBROUTINE DATA
                                .IDENT  /V05.00/

                        ;
                        ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
                        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                        ;
                        ;
                        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                        ; TRANSFERRED.
                        ;
                        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                        ; CORPORATION.
                        ;
                        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                        ;
                        ;
                        ; MODULE DESCRIPTION:
                        ;
                        ; VNP - HELP SUBROUTINE DATA
                        ;
                        ;
                        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                        ;
                        ; IDENT HISTORY:
                        ;
                        ; 1.00  14-DEC-79
                        ;       DECNET-11M/S V3.0
                        ;       DECNET-11M-PLUS V1.0
                        ;
                        ; 3.00  16-APR-82
                        ;       DECNET-11M V3.1
                        ;       DECNET-11M-PLUS V1.1
                        ;
                        ; 4.00  07-NOV-83
                        ;       DECNET-11M V4.0
                        ;       DECNET-11M-PLUS V2.0
                        ;
                        ; 5.00  22-JUL-85
                        ;       DECnet-11M/S V4.2
                        ;       DECnet-11M-Plus V3.0
                        ;       DECnet-Micro/RSX V1.0
                        ;
                        ;
                        ;


```

```
VHELP

SYMBOL

SYMBOL

LO.VER
LR.ACC
LR.ACO
LR.ADD
LR.ALI
LR.BLK
LR.CTL
LR.DES
LR.HLP
LR.LIN
LR.NAM
LR.PAS
LR.PRO
LR.STS
LR.TRI
LR.TYP
LR.UCB
LR.UIC
LR.UNT
LS.ALI
LS.CEX
LS.CIR
LS.CXC
LS.CX5
LS.ECH
LS.FDX
LS.HDX
LS.HLP
LS.LIN
LS.LMO
LS.LOO
LS.MOD
LS.NOD
LS.NTI
LS.OBJ
LS.OPT
LS.PRO
LS.TOP
LS.TST
LS.UNI
LS.XIT
LX.AL
LX.CE
LX.CI
LX.LI
LX.LO
LX.NO
LX.OB
LX.PR
OF$IN
OF$LO
OF$OF
```

```
VHELP      CREATED BY  MACRO  ON 29-JUN-85 AT 02:42     PAGE 2
SYMBOL  CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE              REFERENCES

LO.VER      000042          #5-68
LR.ACC      000020          #5-68
LR.ACO      000052          #5-68
LR.ADD      000012          #5-68
LR.ALI      000002          #5-68
LR.BLK      000010          #5-68
LR.CTL      000004          #5-68
LR.DES      000010          #5-68
LR.HLP      000002          #5-68        6-84
LR.LIN      000002          #5-68        5-68        5-68        5-68        5-68        5-68        5-68
LR.NAM      000012          #5-68
LR.PAS      000041          #5-68
LR.PRO      000002          #5-68
LR.STS      000000          #5-68
LR.TRI      000006          #5-68
LR.TYP      000002          #5-68
LR.UCB      000016          #5-68
LR.UIC      000020          #5-68
LR.UNT      000005          #5-68
LS.ALI   =  000016          #5-68
LS.CEX   =  000001          #5-68
LS.CIR   =  000033          #5-68
LS.CXO   =  010000          #5-68
LS.CX5   =  000400          #5-68
LS.ECH   =  001000          #5-68
LS.FDX   =  004000          #5-68
LS.HDX   =  010000          #5-68
LS.HLP   =  000012          #5-68
LS.LIN   =  000003          #5-68
LS.LMC   =  000007          #5-68
LS.LOG   =  000020          #5-68
LS.MOD   =  000036          #5-68
LS.NOD   =  000014          #5-68
LS.NTI   =  000013          #5-68
LS.OBJ   =  000022          #5-68
LS.OPT   =  000400          #5-68
LS.PRO   =  000002          #5-68
LS.TOP   =  002000          #5-68
LS.TST   =  000010          #5-68
LS.UNF   =  020000          #5-68
LS.XIT   =  000011          #5-68
LX.ALI   =  000017          #5-68
LX.CEX   =  000004          #5-68
LX.CIR   =  000034          #5-68
LX.LIN   =  000006          #5-68
LX.LOG   =  000021          #5-68
LX.NOD   =  000015          #5-68
LX.OBJ   =  000023          #5-68
LX.PRO   =  000005          #5-68
OF$INS   =  000001          #5-68
OF$LOG   =  000100          #5-68
OF$OFF   =  000002          #5-68
```

```
VIMI
Symbo

Size
Size
Opera

Elaps
SY:VI
```

```
                                                                G 10
VIMI - VNP READ IN CEX IMAGE    MACRO V05.03b Saturday 29-Jun-85 02:42  Page 5-7
Symbol table

Size of work file: 11098 Words  ( 44 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:15.26
SY:VIMI.V2,[132,134]VIMI/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VIMI
```

```
VINI -
Table

   5
   6
  14
```

```
H 10
```

Table of contents

```
     5-   54     LOCAL DATA
     6-   60     CONSTANT DEFINITIONS
    14- 1009     GTDUIC - GET DEFAULT UIC
```

```
 002450  000730                          BR      12$                     ;  AND CONTINUE

                                 ;
                                 ; error text
                                 ;

 002452                                  ERMSG$  <Executive symbol missing from symbol table ( %r ).>
 002534                                  ERBLK$  ERR6,<$BFR,2>
 002544                                  ERMSG$  <Commexec symbol missing from symbol table ( %r ).>
 002625                                  ERBLK$  ERR7,<$BFR,2>

                                 ;
                                 ; LIST OF NON-FATAL UNDEFINED SYMBOLS
                                 ;

 002634  052623  021424          OKLIST: .RAD50  'M$$EXT'        ; 11/70 EXTENDED MEMORY
 002640  126025  070704                  .RAD50  '$UMRHD'        ; FIRST UMR ADDRESS
 002644  124545  102532                  .RAD50  '$DEUMR'        ; FIRST FREE UMR
 002650  015343  125456                  .RAD50  'DLC$OV'        ; DLC PROTOCOL OVERHEAD
 002654  012623  125456                  .RAD50  'CRC$OV'        ; CRC OVERHEAD

                                         .IF     DF,R$$MPL
                                         .RAD50  'M$$PRO'        ; MULTI-PROCESSOR SUPPORT
                                         .RAD50  'K$$DAS'        ; KERNAL DATA SPACE SUPPORT
                                         .RAD50  '$SCMOF'        ; STARTING PHYSICAL ADDRESS OF D-SPACE   ;[MP01]
                                         .RAD50  '$CPPAR'        ; POINTER TO PCB OF CPU PARTITION
                                         .RAD50  '$K6TAB'        ; TABLE OF POINTERS TO EACH PROCESSOR'S AREA
                                         .RAD50  '$CPURM'        ; CPU URM MAPPING TABLE POINTER
                                         .RAD50  '$URMTB'        ; UNIBUS RUN CONNECTIVITY TABLE          ;[MP01]
                                         .RAD50  '$POLHD'        ; SECONDARY POOL LISTHEAD                ;[MP01]
                                         .RAD50  '$SECFR'        ; SECONDARY POOL BLOCKS FREE             ;[MP01]
                                         .ENDC

 002660  000000                          .WORD   0

                                 ;
                                 ;       DEFINE SYMBOL BLOCK OFFSETS
                                 ;

 000000                                  .ASECT
         000000                          .=0
 000000                          SB.LNK: .BLKW   1       ; LINK WORD
 000002                          SB.DEF: .BLKW   1       ; FLAG ( NON-ZERO MEANS DEFINED )
 000004                          SB.SYB: .BLKW   2       ; RAD50 SYMBOL NAME
 000010                          SB.VAL: .BLKW   1       ; SYMBOL VALUE
 002662                                  .PSECT
```

```
VINI - VNP INITIALIZATION       MACRO V05.03b Saturday 29-Jun-85 02:43  Page 14-3
Symbol table

Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:39.31
SY:VINI.V2,[132,134]VINI/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VINI
```

```
VLBIO       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 2
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME       REFERENCES

CALL              6-100       6-102       7-121       7-123
GETAD            #5-59        6-100
GETRV            #5-59        6-102
PUTAD            #5-59        7-121
PUTRC            #5-59        7-123
RAND             #6-100       6-100      #6-102       6-102      #7-121       7-121      #7-123       7-123
RETURN            6-104       7-125
SWTCHI           #5-59        6-97        6-103
SWTCHO           #5-59        7-118       7-124
```

```
001400  110022                          MOVB    R0,(R2)+        ; Store field length
001402  001403                          BEQ     20$             ; If eq, no field specified
001404  112122                  10$:    MOVB    (R1)+,(R2)+     ; Else, store password field
001406  005300                          DEC     R0              ;       ...
001410  003375                          BGT     10$             ;       ...
001412                          20$:    RETURN
001414                          101$:   ERROR$  PSWFTL          ; Password field too large


                                ;
                                ; DTE address for SHOW MOD X25-P
                                ;
001434  012700  000020          $PDT::  MOV     #MXDTEL,R0      ; Max DTE length.

001440                          10$:;   IF parsed string length <= max
001440  026700  000000G                   CMP   .PSTCN,R0
001444  003037                            BGT   11$
                                ;       THEN pack into BCD buffer.
001446                                    SAVRG <R3,R4,R5>
001454  005200                            INC   R0              ; Buf size is
001456  006200                            ASR   R0              ;  (max+1)/2.
001460  016701  000000G                   MOV   .PSTPT,R1       ; Ptr to string.
001464  016702  000000G                   MOV   .PSTCN,R2       ; String length.
001470  012705  000047G                   MOV   #RQB+LO.DTE,R5  ; Ptr to buffer.
001474                                    CALL  PCKBCD
001500                                    RESRG <R5,R4,R3>

001506                          20$:;     IF successful
001506  103405                              BCS 21$
                                ;         THEN stash #digits specified and return
001510  016767  000000G 000046G             MOV .PSTCN,RQB+LO.LDT
001516                                      RETURN
001520  000410                            BR    29$
001522                          21$:;     ELSE complain
001522                                      ERROR$  IVDTEA      ; Found a non-digit in the string.
001542                          29$:;     ENDIF

001542  000410                          BR      19$
001544                          11$:;   ELSE complain
001544                                    ERROR$  DTEATL        ; String too long for a DTE address.
001564                          19$:;   ENDIF

001564  062716  000002                  ADD     #2,(SP)
001570                                  RETURN


                                ;
                                ; Destination node user id
                                ;
001572                          $DUSE::
001572  016700  000000G                 MOV     .PSTCN,R0       ; Get user id field length
001576  022700  000020                  CMP     #16.,R0         ; Is user id field too big ?
001602  103412                          BLO     101$            ; If lo, yes
001604  016701  000000G                 MOV     .PSTPT,R1       ; Get start address of user id field
001610  012702  000020G                 MOV     #RQB+LR.UIC,R2  ; Get address of user id buffer
001614  110022                          MOVB    R0,(R2)+        ; Store field length
001616  001403                          BEQ     20$             ; If eq, no field specified
001620  112122                  10$:    MOVB    (R1)+,(R2)+     ; Else, store user id field
001622  005300                          DEC     R0              ;       ...
```

```
                                        ;
                                        ; Unit number
                                        ;
                                                .ENABL  LSB
005522  105767  000001G                 $UNT::  TSTB    $QUAL+1         ; Any wild cards specified ?
005526  001052                                  BNE     ERRLID          ; If ne, yes .. illegal line id
005530  005767  000000G                         TST     .PNUMH          ; Number must be a byte value
005534  001027                                  BNE     101$            ;       ...
005536  105767  000001G                         TSTB    .PNUMB+1        ;       ...
005542  001024                                  BNE     101$            ;       ...
005544                                          CALL    CHKLOG          ; Check for logging function code
005550  103004                                  BCC     10$             ; Br if logging function
005552  116767  000000G 000005G                 MOVB    .PNUMB,RQB+LR.UNT ; Set unit number
005560  000403                                  BR      $XUNT           ; And continue
005562  116767  000000G 000033G 10$:    MOVB    .PNUMB,RQB+LO.UNT ; Set unit number
005570  004267  000000G                 $XUNT:: JSR     R2,$SAVVR       ; Save R0-R2
005574                                          CALL    GETLIN          ; Get SLT and PDV address of line
005600  103415                                  BCS     102$            ; Br if not in system
005602  032761  000000G 000000G                 BIT     #ZF.MUX,Z.FLG(R1) ; Multiplex device?
005610  001421                                  BEQ     ERRLID          ; Br if no
005612                                          RETURN
005614                                  101$:   ERROR$  UNUMTL
005634                                  102$:   ERROR$  NSLINE          ; Line not in system
                                                .DSABL  LSB

005654                                  ERRLID: ERROR$  IVLIID          ; Invalid line identification


                                        ;
                                        ; Vector value
                                        ;
005674  005767  000000G                 $VEC::  TST     .PNUMH          ; Must be single precision
005700  001014                                  BNE     101$            ;       ...
005702  022767  001000  000000G                 CMP     #1000,.PNUMB    ; Less than 1000 ?
005710  103410                                  BLO     101$            ; If lo, no
005712  032767  000003  000000G                 BIT     #3,.PNUMB       ; Is it a multiple of 4 ?
005720  001004                                  BNE     101$            ; If ne, no
005722  016767  000000G 000014G                 MOV     .PNUMB,RQB+LO.VCT ; Set vector value
005730                                          RETURN
005732                                  101$:   ERROR$  ILLVEC

                                        ;
                                        ; Missing tributary or unit number
                                        ;
005752  032767  000000G 000000G $XMISS::BIT     #FL.CNT,$FLAGS  ; Controller only specified?
005760  001423                                  BEQ     10$             ; Br if no
005762  042767  000000G 000000G                 BIC     #FL.CNT,$FLAGS  ; Reset flag
005770  004267  000000G                         JSR     R2,$SAVVR       ; Save R0-R2
005774                                          CALL    GETLIN          ; Get SLT and PDV address of line
006000  103414                                  BCS     101$            ; Br if not in system
006002  032761  000000G 000000G                 BIT     #ZF.MUX,Z.FLG(R1) ; Mux device?
006010  001321                                  BNE     ERRLID          ; Br if yes
006012  032762  000000G 000000G                 BIT     #LF.MTP,L.FLG(R2) ; Multipoint device?
006020  001403                                  BEQ     10$             ; Br [illegible]
006022  105767  000001G                         TSTB    $OPTON+1        ; Necessary to specify tributary
006026  001312                                  BNE     ERRLID          ; Br if yes
006030                                  10$:    RETURN                  ; Finished
```

CKPVC1 - Check for unique PVC name [old format]

```
   499                                          .SBTTL  CKPVC1 - Check for unique PVC name [old format]                   ;[TD001]
   500                                  ;+                                                                                ;[TD001]
   501                                  ;                                                                                 ;[TD001]
   502                                  ; *** CKPVC1 - Check for unique PVC name [old format]                             ;[TD001]
   503                                  ;                                                                                 ;[TD001]
   504                                  ; INPUTS:                                                                         ;[TD001]
   505                                  ;       NONE                                                                      ;[TD001]
   506                                  ;                                                                                 ;[TD001]
   507                                  ; OUTPUTS:                                                                        ;[TD001]
   508                                  ;       CARRY CLEAR - SPECIFIED PVC IS UNIQUE                                     ;[TD001]
   509                                  ;       CARRY SET - SPECIFIED PVC IS NOT UNIQUE                                   ;[TD001]
   510                                  ;       All registers preserved                                                   ;[TD001]
   511                                  ;                                                                                 ;[TD001]
   512                                  ;-                                                                                ;[TD001]
   513 001022                           CKPVC1:                                                                           ;[TD001]
   514 001022                                   CALL    $SAVAL          ; Save all registers                              ;[TD001]
   515 001026  016700  000000G                  MOV     $PSIPT,R0       ; Point to PSI home block                         ;[TD001]
   516 001032  062700  000006                   ADD     #H$PVC,R0       ; Point to PVC name block listhead                ;[TD001]
   517 001036                                   GETRV   R0,#2           ; Read in listhead                                ;[TD001]
   518 001054                                   SAVMAP                  ; Save current mapping                            ;[TD001]
   519 001060  016700  000000G          10$:    MOV     $BFR,R0         ; Get address of next block in list               ;[TD001]
   520 001064  000241                           CLC                     ; Assume success                                  ;[TD001]
   521 001066  001426                           BEQ     30$             ; Branch if end of list                           ;[TD001]
   522 001070                                   CEACC$  R0              ; Convert to mapped address                       ;[TD001]
   523 001100                                   GETAD   R0,#C$LEN       ; Read in PVC name block                          ;[TD001]
   524 001116  012701  000006                   MOV     #PVNAMX,R1      ; Get length of PVC name                          ;[TD001]
   525 001122  012700  000027'                  MOV     #PVCNAM,R0      ; Address of PVC sought                           ;[TD001]
   526 001126  012702  000002G                  MOV     #$BFR+C$NAM,R2  ; Address of candidate                            ;[TD001]
   527 001132  122022                   20$:    CMPB    (R0)+,(R2)+     ; Check if names match                            ;[TD001]
   528 001134  001351                           BNE     10$             ; Branch if no match                              ;[TD001]
   529 001136  005301                           DEC     R1                                                                ;[TD001]
   530 001140  003374                           BGT     20$                                                               ;[TD001]
   531 001142  000261                           SEC                     ; Else error - set c-bit                          ;[TD001]
   532 001144                           30$:    RESMAP                  ; Restore previous mapping                        ;[TD001]
   533 001150                                   RETURN                                                                    ;[TD001]
```

```
VFPPVC      CREATED BY  MACRO  ON 29-JUN-85 AT 02:38      PAGE 4
MACRO CROSS REFERENCE                                      CREF   04.00

MACRO NAME      REFERENCES

BIAS            #5-89
CALL             9-175     11-284    12-328    12-340    12-356    12-372    13-384    13-391    13-393    13-399
                 13-401    14-431    14-435    14-440    14-441    14-443    14-445    14-446    14-452    14-455
                 14-458    15-476    15-479    15-484    15-485    16-514    16-517    16-522    16-523    17-549
                 17-554    17-558    17-563    17-570    17-574    17-581    18-601    18-605    18-609    18-613
                 18-618    18-622    18-629    19-655    19-656    19-659    20-695    20-700    20-706    20-723
                 21-746    21-757    21-768    23-806    23-811
CEACC$          #5-61      15-484    16-522    17-558    18-609    19-656    20-695
DBGTP$          #10-190   #10-203   #10-219   #10-223   #10-227   #10-231   #10-235   #10-239   #10-243   #10-247
                #10-251   #10-255
EMSG$R          #5-61       9-184    11-294    11-310    11-322    12-335    12-347    12-362    12-377    13-409
                 13-410    23-819
GETAD           #5-61      14-441    14-452    15-485    16-523    19-659    20-700    21-757
GETRV           #5-62      14-435    15-479    16-517    19-655
ISTAT$          #5-62      10-190
MAP             #5-83
MTRAN$          #10-190
NTLER$          #5-62       8-145     8-146     8-147     8-148
PHBDF$          #5-62       5-67
PUTAD           #5-61      14-443    14-458    17-570    18-618    20-723    21-768
PVCDF$          #5-61       5-68
RAND            #14-435    14-435   #14-441    14-441   #14-443    14-443   #14-452    14-452   #14-458    14-458
                #15-479    15-479   #15-485    15-485   #16-517    16-517   #16-523    16-523   #17-570    17-570
                #18-618    18-618   #19-655    19-655   #19-659    19-659   #20-700    20-700   #20-723    20-723
                #21-757    21-757   #21-768    21-768
RESMAP          #5-77      14-444    15-496    16-532    17-584    18-632    19-674    21-769
RESRG           #5-61      19-675
RETURN           9-179     11-290    11-306    11-318    12-331    12-343    12-358    12-373    13-405    14-459
                 15-497    16-533    17-585    18-633    19-676    20-724    21-770    22-790    23-815    23-831
SAVMAP          #5-73      14-436    15-480    16-518    17-550    18-602    19-652    21-744
SAVRG           #5-61      19-651
STATE$          #5-62      10-195   #10-198   #10-201   #10-205   #10-208   #10-211   #10-214   #10-217   #10-221
                #10-225   #10-229   #10-233   #10-237   #10-241   #10-245   #10-249   #10-253   #10-259   #10-262
                #10-265   #10-268   #10-271   #10-274
TRAN$           #5-62     #10-196   #10-199   #10-202   #10-203   #10-206   #10-209   #10-212   #10-215   #10-218
                #10-219   #10-222   #10-223   #10-226   #10-227   #10-230   #10-231   #10-234   #10-235   #10-238
                #10-239   #10-242   #10-243   #10-246   #10-247   #10-250   #10-251   #10-254   #10-255   #10-260
                #10-263   #10-266   #10-269   #10-272
VNPDBG          #5-63       9-167
XCBDF$          #5-61       5-69
```

```
VFPRDT -
Symbol t

A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
CERR  =
CFGBF =
CFGSZ =
CFLIN =
CHKRDT
COUNT =
C$$CKP=
C$$ORE=
C$$RSH=
D$$BUG=
D$$ISK=
D$$L11=
D$$YNC=
D$$YNM=
END
E$$XPR=
FMT10 =
FMT8  =
FM.10 =
FM.8  =
F$$LVL=
F.RSIZ=
G$$TPP=
G$$TSS=
G$$TTK=
G$$WRD=
HF$DLM=
HF$GWY=
HF$HOS=

. ABS.

DATA
$STATE
$KTAB
$KSTR
Errors

*** Ass

Work f
Work f
Size of
Size of
Operati

Elapsed
SY:VFPR
```

```
                                                              H 3
VFPRDT - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:39  Page 13-1
Symbol table

A$$CHK= 000000             HF$XDF= 000020             K$$LDC= 000000             RDTENO  000166R            $BLANK= 000006
A$$CPS= 000000             H$CUG   000010             K$$TPS= 000074             RDTENX  000020R       003  $CEACC= ****** GX
A$$PRI= 000000             H$DST   000012             LD$LP  = 000000            RDTEP   000025R       002  $DIGIT= 000024
A$$TRP= 000000             H$D29   000014             LNKEND= ****** GX          RDTEST  000130R            $DNUMB= 000014
CERR   = ****** GX         H$FLG   000000             L$$ASG= 000000             RDTKW   000000RG      004  $EOS   = 000012
CFGBF  = ****** GX         H$GLEN  000104             L$$DRV= 000000             RDTLN  = 000017            $ERRL0  000036R       002
CFGSZ  = ****** GX         H$GLT   000044             L$$P11= 000001             RDTNAM  000000R       002  $ERRL1  000066R       002
CFLIN  = ****** GX         H$GNAM  000050             L$$11R= 000000             RDTST   000000RG      003  $ERRL2  000122R       002
CHKRDT  000450R            H$GNML= 000020             M$$CRB= 000124             RTSPC  = ****** GX         $ERR1T= ****** GX
COUNT  = ****** GX         H$GPT   000046             M$$CRX= 000000             R$DAL   000013             $EXIT  = 000000
C$$CKP= 000000             H$HITS  000034             M$$FCS= 000000             R$DTE   000014             $FAIL  = 177777
C$$ORE= 000400             H$HLEN  000044             M$$MGE= 000000             R$LEN   000024             $GETAD= ****** GX
C$$RSH= 177564             H$LBDA  000070             M$$NET= 000000             R$LNK   000000             $GETRV= ****** GX
D$$BUG= 177514             H$LBDN  000072             M$$OVR= 000000             R$NAM   000002             $LAMDA= 000000
D$$ISK= 000000             H$LDTE  000002             NEXT   = ****** GX         R$OWN   000010             $NUMBR= 000002
D$$L11= 000001             H$LEN   000042             N$$ACC= 000001             R$$DER= 000000             $PSIPT= ****** GX
D$$YNC= 000000             H$LOTS  000032             N$$BUF= 000001             R$$K11= 000001             $PUTAD= ****** GX
D$$YNM= 000000             H$NETW  000024             N$$LDV= 000001             R$$SND= 000000             $QRDT   000000RG
END     000036R       003  H$NML  = 000006            N$$MCP= 000001             R$$11M= 000000             $RADDR= ****** GX
E$$XPR= 000000             H$NPT   000022             N$$MLL= 000001             SPACE  = 000040            $RAD50= 000016
FMT10  = ****** GX         H$PTB   000020             N$$MOV= 000010             STRNXT= ****** GX          $RBLCK= ****** GX
FMT8   = ****** GX         H$PVC   000006             N$$NCT= 000001             SYNERR= ****** GX          $SAVAL= ****** GX
FM.10  = 000000            H$RDTE  000004             N$$PEM= 000001             SYSFDB= ****** GX          $SAVRG= ****** GX
FM.8   = 000000            H$RNW   000042             PCKBCD= ****** GX          S$$WRG= 000000             $STRNG= 000004
F$$LVL= 000001             H$SVC   000036             P$$P45= 000000             S$$YSZ= 007600             $SUBXP= 000010
F.RSIZ= ****** GX          H$TRB   000016             P$$WRD= 000000             T$$KMG= 000000             $$$FLG= 177777
G$$TPP= 000000             H$XAVL  000100             Q$$OPT= 000010             T$$MIN= 000000             $$$KEY= 000000
G$$TSS= 000000             H$XBIA  000074             RDEND   000212R            V$$CTR= 001000             $$$R   = 000010
G$$TTK= 000000             H$X29C  000040             RDNAM   000052R            X$$DBT= 000000             $$$STA= 000000
G$$WRD= 000000             I$$RAR= 000000             RDTDF   000000R       003  $ALPHA= 000022             $$$TMP= 000000R       005
HF$DLM= 000002             I$$RDN= 000000             RDTE    000006R       002  $ALPOL= ****** GX          .PSTCN= ****** GX
HF$GWY= 000010             K$$CNT= 177546             RDTEMX= 000006             $ANY   = 000020            .PSTPT= ****** GX
HF$HOS= 000004             K$$CSR= 177546             RDTEN  = 000024R      002  $BFR   = ****** GX         .TPARS= ****** GX

. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        000604    001   (RW,I,LCL,REL,CON)
DATA    000176    002   (RW,D,LCL,REL,CON)
$STATE  000046    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13495 Words  ( 53 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:23.58
SY:VFPRDT.V2,[132,134]VFPRDT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPRDT
```

```
                                          .SBTTL  Parse table for RNA$DF

000062                                    STATE$  RNADF
000062                                      TRAN$   !SKIPSP                 ; Skip leading spaces
000062                                    STATE$
000062                                      TRAN$   %RNA$DF%,,,1,SYNERR     ; Flag RNA$DF recognised

000062                                    STATE$
000062                                      TRAN$   !SKIPSP
000062                                    STATE$
000062                                      TRAN$   !END, $EXIT             ; Check for end
000062                                      TRAN$   !STRING,,RNAUSR         ; Parse user name

000062                                    STATE$
000062                                      TRAN$   !END, $EXIT             ; Check for end
000062                                      TRAN$   !OPSEP                  ; Skip operand separator
000062                                    STATE$
000062                                      TRAN$   !STRING,,RNAPAS         ; Parse password

000062                                    STATE$
000062                                      TRAN$   !END, $EXIT             ; Check for end
000062                                      TRAN$   !OPSEP                  ; Skip operand separator
000062                                    STATE$
000062                                      TRAN$   !STRING,,RNAACT         ; Parse account

000062                                    STATE$
000062                                      TRAN$   !END,$EXIT              ; Ensure correct termination
000062                                    STATE$
```

```
                                ;+
                                ; *** STORCH - Store string character
                                ;
                                ; INPUTS
                                ;       STRLEN - Number of characters read so far
                                ;       .PCHAR - Current character to be stored
                                ;
                                ; OUTPUTS
                                ;       STRLEN incremented
                                ;       If chars read <= max string size, character stored in string buffer
                                ;-
001020  126727  000164' 000020  STORCH: CMPB    STRLEN,#MAXSTR          ; Check string not too long
001026  103005                          BHIS    10$                     ; Do not store character if no room
001030  116700  000164'                 MOVB    STRLEN,R0
001034  116760  000000G 000165'         MOVB    .PCHAR,STRBUF(R0)       ; Store character in buffer
001042  105267  000164'         10$:    INCB    STRLEN                  ; Count this character
001046                                  RETURN
        000001                          .END
```

```
                                 .SBTTL  TPARS STATE TABLES
                         ;
                         ; TPARS STATE TABLES
                         ;
000052                           ISTAT$  X29ST,X29KW
                         ;
                         ; X29$DF
                         ;
000052                           STATE$  X29DF
000052                            TRAN$  %X29$DF%,,,1,SYNERR

000052                           STATE$
000052                            TRAN$  $NUMBR,,X29MXC          ; MAXIMUM CIRCUITS

000052                           STATE$
000052                            TRAN$  <',>

000052                           STATE$
000052                            TRAN$  $NUMBR,,X29CTM          ; COUNTER TIMER

000052                           STATE$
000052                            TRAN$  !END,$EXIT,X29END
                         ;
                         ; CHECK FOR END OF SOURCE LINE
                         ;
000052                           STATE$  END
000052                            TRAN$   <';>,$EXIT
000052                            TRAN$   $EOS,$EXIT

000052                           STATE$
```

```
                                        .SBTTL  LOOK FOR ***$DF MACRO

                                ;+
                                ; Q*** - LOOK FOR ***$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3,R4,R5=DESTROYED
                                ;-
                                        .ENABL  LSB

000000  012705  000000'         $QX3P:: MOV     #X3PDF,R5       ; STATE TABLE ADDRESS
000004  000402                          BR      10$
000006  012705  000036'         $QCHN:: MOV     #CHNDF,R5       ; STATE TABLE ADDRESS
                                ;
000012  005001                  10$:    CLR     R1              ; FULL KEYWORD MATCH LENGTH
000014  012702  000000'                 MOV     #X3PKW,R2       ; KEYWORD TABLE ADDRESS
000020  016703  000000G                 MOV     CFGSZ,R3        ; RECORD LENGTH
000024  012704  000000G                 MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000030  005067  000000G                 CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000034                                  CALL    .TPARS          ; GO DO THE PARSE
000040  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000042  005367  000000G                 DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000046  001401                          BEQ     101$            ; IF EQ, YES
000050                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000052                          101$:   EMSG$R  1T              ; SYNTAX ERROR

                                        .DSABL  LSB
```

```
     55                                          ;****
     56                                          ; MACRO CALLS
     57                                          ;****
     58
     59                                                  .MCALL  HLPDF$
     60
     61                                          ;****
     62                                          ; LOCAL DATA
     63                                          ;****
     64
     65                                          ;
     66                                          ; HELP FILE DEFINITION
     67                                          ;
     68 000000                                           HLPDF$  4,1,1,<VNP>,,DATSET,,<VNP>
     69
     70                                          ;
     71                                          ; DEFINE THE HELP LIBRARY FILE
     72                                          ;
     73 000160  000003  000174' 000005  DATSET: .WORD   DEVL,DEVA,UICL,UICA,FILL,FILA
        000166  000177' 000012  000204'
     74
     75 000174     114     102     072  DEVA:   .ASCII  /LB:/
     76         000003                  DEVL=   .-DEVA
     77
     78 000177     133     061     054  UICA:   .ASCII  /[1,2]/
        000202     062     135
     79         000005                  UICL=   .-UICA
     80
     81 000204     116     105     124  FILA:   .ASCII  /NETHLP.ULB/
        000207     110     114     120
        000212     056     125     114
        000215     102
     82         000012                  FILL=   .-FILA
     83
     84                                         .EVEN
     85
     86
     87         000001                          .END
```

```
VHELP

SYMBOL

SYMBOL

OF$ON
OF$SM
OP$ADD
OP$ALL
OP$CC
OP$CI
OP$CO
OP$CO
OP$CO
OP$DS
OP$DT
OP$DU
OP$ES
OP$EV
OP$FI
OP$GR
OP$HO
OP$HT
OP$IN
OP$IN
OP$KD
OP$KD
OP$KE
OP$KG
OP$KN
OP$KS
OP$LC
OP$LN
OP$LO
OP$LS
OP$LT
OP$LV
OP$MA
OP$MC
OP$MD
OP$ML
OP$MO
OP$MS
OP$NA
OP$NE
OP$NL
OP$NO
OP$OU
OP$OW
OP$PA
OP$PR
OP$RE
OP$RP
OP$SC
OP$SE
OP$SE
OP$SI
```

```
VHELP      CREATED BY  MACRO  ON 29-JUN-85 AT 02:42      PAGE 3
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE            REFERENCES

OF$ON    = 000000         #5-68
OF$SMC   = 000200         #5-68
OP$ADD   = 000200         #5-68
OP$ALL   = 000200         #5-68
OP$CCS   = 000004         #5-68
OP$CIE   = 004000         #5-68
OP$CON   = 000400         #5-68
OP$COP   = 000001         #5-68
OP$COS   = 000100         #5-68
OP$DST   = 000200         #5-68
OP$DTE   = 000040         #5-68
OP$DUP   = 000002         #5-68
OP$EST   = 000100         #5-68
OP$EVE   = 010000         #5-68
OP$FIL   = 000200         #5-68
OP$GRP   = 000100         #5-68
OP$HOS   = 000001         #5-68
OP$HTM   = 000002         #5-68
OP$INA   = 004000         #5-68
OP$INC   = 001000         #5-68
OP$KDS   = 002000         #5-68
OP$KDT   = 000400         #5-68
OP$KEV   = 001000         #5-68
OP$KGR   = 001000         #5-68
OP$KNT   = 010000         #5-68
OP$KSK   = 002000         #5-68
OP$LCT   = 000200         #5-68
OP$LNE   = 000040         #5-68
OP$LOC   = 000001         #5-68         #5-68
OP$LST   = 000002         #5-68
OP$LTM   = 000001         #5-68
OP$LVL   = 000010         #5-68
OP$MAC   = 000010         #5-68
OP$MCO   = 000002         #5-68
OP$MDE   = 000400         #5-68
OP$MLI   = 000004         #5-68
OP$MON   = 000100         #5-68
OP$MST   = 000020         #5-68
OP$NAM   = 000004         #5-68         #5-68         #5-68
OP$NET   = 004000         #5-68
OP$NLI   = 000002         #5-68
OP$NOD   = 000020         #5-68
OP$OUT   = 002000         #5-68
OP$OWN   = 000020         #5-68
OP$PAR   = 000010         #5-68
OP$PRI   = 000010         #5-68
OP$RET   = 010000         #5-68
OP$RPA   = 000010         #5-68
OP$SCO   = 000001         #5-68
OP$SEG   = 000400         #5-68
OP$SER   = 000400         #5-68
OP$SIN   = 000010         #5-68
```

```
VIMI        CREATED BY  MACRO  ON 29-JUN-85 AT 02:42     PAGE 1
SYMBOL  CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE              REFERENCES

EROUT      002012 R           5-316    5-317   #5-337
ERR1       001266 R           5-315   #5-324
ERR2       001344 R           5-316   #5-326
ERR3       002004 R           5-317   #5-332
FATAL    = ******  GX         5-315    5-316    5-317
FE.EXV   = 000004             5-121
F.RSIZ   = ******  GX        *5-244   *5-284   *5-295
F.URBD   = ******  GX        *5-101   *5-105   *5-172   *5-224   *5-281   *5-288
GCLBUF   = ******  GX         5-332    5-332   *5-342   *5-343
HF$GWY   = 000010             5-300
HF$HOS   = 000004             5-297
HF$XDF   = 000020             5-303
H$FLG      000000             5-294
PF.PSV   = ******  GX         5-302
PF.XDF   = ******  GX         5-305
PF.XGA   = ******  GX         5-299
R$$MPL   = ******             5-87     5-153    5-195    5-277
SYSFDB   = ******  GX        *5-101   *5-105   *5-172   *5-224   *5-244   *5-281   *5-284   *5-288   *5-295
Z.LEN    = ******  GX         5-150
$ADRX    = ******  GX        *5-231   *5-236    5-241
$ADRZ    = ******  GX        *5-230   *5-237    5-240
$BFR     = ******  GX         5-288    5-297    5-300    5-303   *5-311   *5-312    5-313    5-324    5-324
                              5-341    5-342    5-343
$BIGCT   = ******  GX        *5-163    5-169   *5-179   *5-183    5-186   *5-213
$CEAVL   = ******  GX         5-281    5-285
$CETAB   = ******  GX         5-105    5-106    5-132    5-157    5-167
$CETDF   = ******  GX        *5-133
$CETLN   = ******  GX        *5-160
$CETND   = ******  GX         5-106
$CETST   = ******  GX        *5-115
$CEXND   = ******  GX         5-102
$CLBUF   = ******  GX        *5-341
$DBG$    = ******             5-99
$EROUT   = ******  GX         5-315    5-344
$FMASK   = ******  GX         5-121
$GETAD   = ******  GX         5-244
$GETRV   = ******  GX         5-103    5-127    5-185    5-284    5-295
$LLCTA   = ******  GX        *5-136    5-249    5-271
$NTLHB   = ******  GX         5-224
$NTLPT   = ******  GX         5-241
$PDVNM   = ******  GX         5-141    5-270
$PDVTA   = ******  GX         5-101    5-102    5-114    5-115    5-127    5-131   *5-135    5-140    5-269
$PFLAG   = ******  GX        *5-296   *5-299   *5-302   *5-305
$PSIPT   = ******  GX         5-292
$RADDR   = ******  GX        *5-103   *5-127   *5-173    5-230    5-236   *5-282   *5-283   *5-295
$RBLCK   = ******  GX         5-231    5-237
$RSIZE   = ******  GX        *5-102   *5-107   *5-110    5-116   *5-126    5-161    5-168    5-171    5-173
                              5-174   *5-178   *5-182
$SAVAL   = ******  GX         5-100
$SLTMA   = ******  GX        *5-137    5-214    5-263    5-265
$SLTNM   = ******  GX         5-215    5-250    5-264
$UNBSZ   = ******  GX        *5-232   *5-238
```

```
		.TITLE	VINI - VNP INITIALIZATION
		.IDENT	/V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;	VNP - TASK INITIALIZATION
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	27-FEB-78
;	VERSION 2.0 RELEASE
;
; 2.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;

```

```
                                ;+
                                ;       **-VSYM1
                                ;
                                ;       READ SYSTEM IMAGE FILE FOR WORD VALUES THAT ARE NOT DEFINED
                                ;       FROM SYMBOL TABLE. UPDATE BOUNDARY CONDITIONS FOR RANDOM ACCESS
                                ;       I/O.
                                ;
                                ; INPUTS:
                                ;       SYSTEM IMAGE FILE OPEN
                                ;
                                ; OUPUTS:
                                ;       UPDATED SYMBOL TABLE
                                ;       BOUNDARY CONDITIONS FOR GETRC,GETRV,PUTRC,GETAD SATISFIED
                                ;
                                ;-

002662  012702  000000G         VSYM1:  MOV     #$BFR,R2         ; POINT AT I/O BUFFER
002666                                  GETRV   .SYSIZ,#6        ; READ IN THREE WORDS
002706  011267  000000G                 MOV     (R2),$SYSIZ      ; SIZE OF SYSTEM IN 32-WORD BLOCKS

                                        .IF NDF R$$MPL                                          ;[MP01]
002712  016267  000002  000000G         MOV     2(R2),$VCTVB     ; VBN OF DISPLACED MEMORY
002720  016267  000004  000000G         MOV     4(R2),$VCTOF     ; OFFSET OF VIRTUAL ZERO WITHIN BLOCK
                                        .ENDC                                                   ;[MP01]

                                ; USE AS MUCH OF THE SYSTEM IMAGE FILE AS IS ALLOCATED

002726  016703  000000G                 MOV     $SYSIZ,R3        ; CALCULATE NUMBER OF
002732  000241                          CLC                      ; CLEAR C-BIT
002734  006003                          ROR     R3               ;  DISK BLOCKS
002736  006203                          ASR     R3               ;   NEEDED TO CONTAIN
002740  006203                          ASR     R3               ;    SYSTEM IMAGE ( 32. X 8. = 256 . )
002742  062703  000002                  ADD     #2,R3            ; LABEL GROUP IS 2 BLOCKS
002746  005067  000014G                 CLR     F.FFBY+SYSFDB    ; FIRST FREE BYTE IS 0
002752  005203                          INC     R3               ; AND THEREFORE EOFBN MUST BE END+1
002754  020367  000006G                 CMP     R3,F.HIBK+2+SYSFDB ; IS FILE BIG ENOUGH ?
002760  101403                          BLOS    5$               ; YES
002762  016703  000006G                 MOV     F.HIBK+2+SYSFDB,R3 ; USE LARGEST NUMBER POSSIBLE
002766  005203                          INC     R3               ; ADJUST FOR FIRST FREE BYTE
002770  020367  000012G         5$:     CMP     R3,F.EFBK+2+SYSFDB ; IS EOF BLOCK SMALLER THAN NECESSARY ?
002774  101402                          BLOS    8$               ; NO
002776  010367  000012G                 MOV     R3,F.EFBK+2+SYSFDB ; SET EOF BLOCK NUMBER ( LOW PART )

003002                          8$:     GETRV   .FMASK,#2        ; READ SYSTEM FEATURE MASK WORD ( $FMASK )
003022  011267  000000G                 MOV     (R2),$FMASK      ; GET $FMASK VALUE IN LOCAL STORAGE
003026                                  GETRV   .EXSIZ           ; READ EXEC SIZE (THIS IS AN ADDRESS)
003040  011267  000000G                 MOV     (R2),$EXSIZ      ; UPDATE EXEC'S HIGHEST ADDRESS IN ROOT
003044  022712  100000                  CMP     #100000,(R2)     ; LEGAL SIZE
003050  032767  000004  000000G         BIT     #FE.EXV,$FMASK   ; 20K EXEC ?
003056  001402                          BEQ     10$              ; NOT 20K EXEC
003060  022712  120000                  CMP     #120000,(R2)     ; LEGAL SIZE ?
003064  103476                  10$:    BCS     110$             ; ILLEGAL EXEC SIZE
003066                                  GETRV   .DEVHD           ; DEVICE LISTHEAD
003100  011267  000000G                 MOV     (R2),$DEVHD      ; UPDATE ROOT SEGEMENT VALUE
003104                                  GETRV   .PARHD           ; PARTITION LISTHEAD
003116  011267  000000G                 MOV     (R2),$PARHD      ; UPDATE ROOT SEGEMENT VALUE
003122                                  SWTCHI  #$CRAVL-2
```

```
VINI       CREATED BY  MACRO  ON 29-JUN-85 AT 02:43       PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL    VALUE              REFERENCES

ABTIM       003726 R          12-883      #12-886
CEAVL       004060 R          12-917      #12-920
CEPWR       004116 R          12-930      #12-933
CEXSTB      004046 R          9-430        10-503      #12-914
CLINS       003430 R          12-826      #12-829
CLKHD       003752 R          12-889      #12-892
CL.AST   =  004000           #8-138
CL.ATT   =  000200           #8-138
CL.CAS   =  020000           #8-138
CL.CMT   =  000001           #8-138       8-144       8-151
CL.EIF   =  000040           #8-138
CL.EXN   =  040000           #8-138
CL.EXT   =  000010           #8-138
CL.IND   =  000002           #8-138
CL.LCC   =  000020           #8-138
CL.NPR   =  000400           #8-138
CL.OPN   =  000100           #8-138
CL.RNE   =  000004           #8-138
CL.XIT   =  100000           #8-138       8-154
CRAVL       003310 R          12-702      #12-805
CRCOV       004104 R          12-927      #12-930
CS.OUT   =  000002            8-164
C.CMLD   =  000002           *8-162      *8-162
C.SIZE   =  000054            8-162       8-164
DEFDEV      000000 R         #5-56        9-403
DEUMR       003702 R          12-877      #12-880
DEVHD       003334 R          12-808      #12-811
DLCOV       004072 R          12-920      #12-927
ERROR1      000200 R          7-116       #7-118
ERR0        000724 R         #8-261
ERR1        000770 R          8-249       #8-263
ERR10       004432 R          13-986      #13-1000
ERR11       004520 R          13-988      #13-1002
ERR12       004602 R          13-991      #13-1004
ERR13       001310 R          8-254       #8-272
ERR14       004754 R          13-993      #13-1007
ERR2        001060 R          8-251       #8-266
ERR3        001170 R          8-253       #8-269
ERR4        002140 R          9-449       #9-460
ERR5        002256 R          9-450       #9-463
ERR6        002536 R          9-453        10-516     #10-529
ERR7        002626 R          10-519      #10-531
EXSIZ       003346 R          12-811      #12-814
EXSTB       003276 R          9-363        10-480     #12-698
FATAL    =  ******   GX       7-116        8-253       8-254       9-449       9-450       9-453       11-671
FD.INS   =  ******   GX       8-177
FE.CEX   =  020000            7-108
FE.EXV   =  000004            11-617
FILE     =  ******   GX       8-143        8-157       8-224       8-263       8-266       8-269       8-272       9-302       9-460
                              9-463
FMASK       003360 R          12-814      #12-817
FO.MFY   =  ******   GX       8-168
```

```
**FILE**ID**VNCP






VV      VV  NN      NN    CCCCCCCC  PPPPPPPP
VV      VV  NN      NN    CCCCCCCC  PPPPPPPP
VV      VV  NN      NN  CC          PP      PP
VV      VV  NN      NN  CC          PP      PP
VV      VV  NNNN    NN  CC          PP      PP
VV      VV  NNNN    NN  CC          PP      PP
VV      VV  NN  NN  NN  CC          PPPPPPPP
VV      VV  NN  NN  NN  CC          PPPPPPPP
VV      VV  NN    NNNN  CC          PP
VV      VV  NN    NNNN  CC          PP
  VV  VV    NN      NN  CC          PP              ....
  VV  VV    NN      NN  CC          PP              ....
    VV      NN      NN    CCCCCCCC  PP              ....
    VV      NN      NN    CCCCCCCC  PP              ....




LL            SSSSSSSS  TTTTTTTTTT
LL            SSSSSSSS  TTTTTTTTTT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL          SS              TT
LL            SSSSSS        TT
LL            SSSSSS        TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LL                  SS      TT
LLLLLLLLLL  SSSSSSSS        TT
LLLLLLLLLL  SSSSSSSS        TT
```

```
001624  003375                                  BGT     10$             ;       ...
001626                                  20$:    RETURN
001630                                  101$:   ERROR$  UIDTL           ; User id field too large

                                        ;
                                        ; Executor node
                                        ;
001650                                  $EXNOD::
001650                                          GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK

001670  016767  000014G 000012G                 MOV     $BFR+D$LNUM,RQB+LR.ADD ; Store executor node
001676  052767  000000G 000000G                 BIS     #VF.ADD,$VFLAG  ; Reminder to use address check, not name check
001704                                          RETURN


                                        ;
                                        ; Store group name
                                        ;
001706  012700  000024G                 $GRO::  MOV     #RQB+LO.GRO,R0  ; Buffer ptr                                     ;[TD001]
001712  012701  000006                          MOV     #MXGRPN,R1      ; Max. size of old-format group name             ;[TD001]
001716                                          CALL    STORID          ; Check and store group name                     ;[TD001]
001722  103401                                  BCS     101$            ; Branch if name too long                        ;[TD001]
001724                                          RETURN                                                                   ;[TD001]
001726                                  101$:   ERROR$  NAMETL                                                           ;[TD001]

                                        ;
                                        ; Hello timer
                                        ;
001746                                  $HTM::
001746  005767  000000G                         TST     .PNUMH          ; Must be a word value
001752  001004                                  BNE     101$            ; If ne, illegal
001754  016767  000000G 000032G                 MOV     .PNUMB,RQB+LO.HTM ; Store the timer value.
001762                                  10$:    RETURN
001764                                  101$:   ERROR$  BDHELT          ; Illegal timer value.

                                        ;
                                        ; If node name specified ...
                                        ;
002004  032767  000000G 000000G $IFNNA::BIT     #VF.ADD,$VFLAG  ; Look at node address bit:
002012  001001                                  BNE     99$             ; Set means address was specified.
002014                                          RETURN                  ; Otherwise, we're golden.

002016  062716  000002                  99$:    ADD     #2,(SP)
002022                                          ERROR$  BDGRP           ; BAD PARAM GROUPING.

                                        ;
                                        ; Test whether a particular entity was specified.
                                        ;
                                                .ENABL  LSB
002042  012700  002156'                 $IFLOG::MOV     #IFLOGT,R0      ; LOGGING function codes
002046  000430                                  BR      10$

002050  012700  002152'                 $IFALI::MOV     #IFALIT,R0      ; ALIAS function codes
002054  000425                                  BR      10$

002056  032767  000001  000000G $IFXAC::BIT     #OP$XAC,$OPTON  ; Ck for X25-ACCESS bit.
002064  001427                                  BEQ     99$
```

```
   1280 006032                        101$:    ERROR$  NSLINE          ; Line not in system
   1281
```

```
                                         .SBTTL  ALPVC0 - Allocate PVC name block [new format] & circuit block  ;[TD001]
                                ;+                                                                              ;[TD001]
                                ; *** ALPVC0 - Allocate PVC name block [new format] & circuit block             ;[TD001]
                                ;                                                                               ;[TD001]
                                ; INPUTS                                                                        ;[TD001]
                                ;        R1 - Port number allocated to PVC                                      ;[TD001]
                                ;        All parsed fields for PVC                                              ;[TD001]
                                ;                                                                               ;[TD001]
                                ; OUTPUTS                                                                       ;[TD001]
                                ;        Registers preserved                                                    ;[TD001]
                                ;        Carry clear PVC block linked into chain                                ;[TD001]
                                ;        Carry set if unable to allocate PVC or X25 circuit block               ;[TD001]
                                ;-                                                                              ;[TD001]
001152                          ALPVC0:                                                                         ;[TD001]
001152                                   CALL    $SAVAL                 ; Save registers                        ;[TD001]
001156                                   SAVMAP                         ; Save current mapping                  ;[TD001]
001162  010146                           MOV     R1,-(SP)               ; Save port number                      ;[TD001]
001164  116701  000026'                  MOVB    PVCNML,R1              ; Get length of PVC name                ;[TD001]
001170  062701  000016                   ADD     #C$LEN,R1              ; Compute total length of PVC name block;[TD001]
001174                                   CALL    $ALPOL                 ; Attempt to allocate block             ;[TD001]
001200  012603                           MOV     (SP)+,R3               ; Get port number back                  ;[TD001]
001202  103455                           BCS     50$                    ; Branch if allocation failed           ;[TD001]
001204  010005                           MOV     R0,R5                  ; Save unmapped address of allocation   ;[TD001]
001206                                   CEACC$  R0,R4                  ; Map to allocation                     ;[TD001]
001216  012702  000016G                  MOV     #$BFR+C$GNAM,R2        ; Target for PVC name                   ;[TD001]
001222  012700  000026'                  MOV     #PVCNML,R0             ; Address of PVC name buffer            ;[TD001]
001226  112001                           MOVB    (R0)+,R1               ; Get length of name                    ;[TD001]
001230  110167  000002G                  MOVB    R1,$BFR+C$NML          ; Move name length                      ;[TD001]
001234                                   CALL    MOVE                   ; Move PVC name                         ;[TD001]
001240  116767  000025' 000015G          MOVB    PVCFLG,$BFR+C$FLG      ; Move PVC flags                        ;[TD001]
001246  016767  000004' 000010G          MOV     PVCLCN,$BFR+C$LCN      ; Move channel number                   ;[TD001]
001254  016767  000000G 000012G          MOV     DTEDES,$BFR+C$DTE      ; Move DTE descriptor                   ;[TD001]
001262  110367  000014G                  MOVB    R3,$BFR+C$PORT         ; Move port number                      ;[TD001]
001266  116701  000026'                  MOVB    PVCNML,R1              ; Recompute block length                ;[TD001]
001272  062701  000016                   ADD     #C$LEN,R1                                                      ;[TD001]
001276                                   PUTAD   R4,R1                  ; Write out PVC name block              ;[TD001]
                                ;                                                                               ;[TD001]
                                ; Attempt to allocate X25 circuit block                                         ;[TD001]
                                ;                                                                               ;[TD001]
001312                                   CALL    ALLXCB                 ; Allocate block                        ;[TD001]
001316  103407                           BCS     50$                    ; Exit if no store                      ;[TD001]
                                ;                                                                               ;[TD001]
                                ; Link PVC into chain                                                           ;[TD001]
                                ;                                                                               ;[TD001]
001320  016700  000000G                  MOV     $PSIPT,R0              ; Form address of listhead              ;[TD001]
001324  062700  000006                   ADD     #H$PVC,R0                                                      ;[TD001]
001330                                   CALL    LNKEND                 ; Link PVC name (address of PVC in R5)  ;[TD001]
                                ;                                                                               ;[TD001]
001334  000241                           CLC                            ; Signal success                        ;[TD001]
001336                          50$:     RESMAP                         ; Restore mapping                       ;[TD001]
001342                                   RETURN                                                                 ;[TD001]
```

```
**FILE**ID**VFPRDT






VV        VV    FFFFFFFFFF      PPPPPPPP        RRRRRRRR        DDDDDDDD        TTTTTTTTTT
VV        VV    FFFFFFFFFF      PPPPPPPP        RRRRRRRR        DDDDDDDD        TTTTTTTTTT
VV        VV    FF              PP        PP    RR        RR    DD        DD        TT
VV        VV    FF              PP        PP    RR        RR    DD        DD        TT
VV        VV    FF              PP        PP    RR        RR    DD        DD        TT
VV        VV    FF              PP        PP    RR        RR    DD        DD        TT
VV        VV    FFFFFFFF        PPPPPPPP        RRRRRRRR        DD        DD        TT
VV        VV    FFFFFFFF        PPPPPPPP        RRRRRRRR        DD        DD        TT
VV        VV    FF              PP              RR  RR          DD        DD        TT
VV        VV    FF              PP              RR  RR          DD        DD        TT
  VV    VV      FF              PP              RR    RR        DD        DD        TT          ....
  VV    VV      FF              PP              RR    RR        DD        DD        TT          ....
    VV          FF              PP              RR        RR    DDDDDDDD            TT          ....
    VV          FF              PP              RR        RR    DDDDDDDD            TT          ....





LL                SSSSSSSS      TTTTTTTTTT
LL                SSSSSSSS      TTTTTTTTTT
LL              SS                  TT
LL              SS                  TT
LL              SS                  TT
LL              SS                  TT
LL                SSSSSS            TT
LL                SSSSSS            TT
LL                        SS        TT
LL                        SS        TT
LL                        SS        TT
LL                        SS        TT
LLLLLLLLLL      SSSSSSSS            TT
LLLLLLLLLL      SSSSSSSS            TT
```

```
VFPRDT

SYMBOL

SYMBOL

CERR
CFGBF
CFGSZ
CFLIN
CHKRDT
COUNT
FMT10
FMT8
FM.10
FM.8
F.RSIZ
H$RDTE
LNKEND
NEXT
N$$VCT
PCKBCD
RDEND
RDNAM
RDTDF
RDTE
RDTEMX
RDTEN
RDTENO
RDTEP
RDTEST
RDTKW
RDTLN
RDTNAM
RDTST
RTSPC
R$DAL
R$DTE
R$LEN
R$NAM
SPACE
STRNXT
SYNERR
SYSFDB
S$$BAS
$ALPHA
$ALPOL
$ANY
$BFR
$BLANK
$CEACC
$DIGIT
$DNUMB
$EOS
$ERRL0
$ERRL1
$ERRL2
$ERR1T
```

```
VFPRDT      CREATED BY  MACRO  ON 29-JUN-85 AT 02:39      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

CERR     = ******  GX         8-104       8-105       8-106
CFGBF    = ******  GX         9-129
CFGSZ    = ******  GX         9-128
CFLIN    = ******  GX         8-104       8-105       8-106
CHKRDT     000450 R          12-232     #13-289
COUNT    = ******  GX       *11-206      12-253
FMT10    = ******  GX         8-104       8-105
FMT8     = ******  GX         8-106
FM.10    = 000000            #8-104      #8-105
FM.8     = 000000            #8-106
F.RSIZ   = ******  GX       *12-261     *13-291     *13-298
H$RDTE     000004            12-263      13-290
LNKEND   = ******  GX        12-264
NEXT     = ******  GX       *11-207      11-214
N$$VCT   = ******            12-241      13-297
PCKBCD   = ******  GX        12-231
RDEND      000212 R         #12-227
RDNAM      000052 R         #11-179
RDTDF      000060 R           9-125
RDTE       000006 R          #7-95       11-202      11-207      12-229
RDTEMX   = 000006            #6-83       11-180      12-249      13-302
RDTEN    = 000024 R          #7-96       11-214
RDTENO     000166 R         #11-214
RDTEP      000025 R          #7-97       12-230      12-255
RDTEST     000130 R         #11-201
RDTKW      000000 RG          9-127     #10-146
RDTLN    = 000017            #7-94        7-95        7-97       11-201      12-228      12-254
RDTNAM     000000 R          #7-93       11-183      11-187      12-248      13-301
RDTST      000000 RG        #10-146
RTSPC    = ******  GX         8-104       8-105       8-106
R$DAL      000013           *12-253
R$DTE      000014            12-257
R$LEN      000024            12-234      12-261      13-298
R$NAM      000002            12-247      13-300
SPACE    = 000040            #6-84       11-184
STRNXT   = ******  GX        11-208      11-216
SYNERR   = ******  GX        *9-130      *9-133
SYSFDB   = ******  GX       *12-261     *13-291     *13-298
S$$BAS   = ******             8-104       8-104       8-105       8-105       8-106       8-106
$ALPHA   = 000022           #10-146
$ALPOL   = ******  GX        12-237
$ANY     = 000020           #10-146
$BFR     = ******  GX        12-242      12-246     *12-253      12-256      13-294      13-299
$BLANK   = 000006           #10-146
$CEACC   = ******  GX        12-241      13-297
$DIGIT   = 000024           #10-146
$DNUMB   = 000014           #10-146
$EOS     = 000012           #10-146
$ERRL0     000036 R          #8-104      11-196      12-271
$ERRL1     000066 R          #8-105      11-221
$ERRL2     000122 R          #8-106      12-272
$ERR1T   = ******  GX         9-140
```

```
                                ;
                                ; Parser subexpressions
                                ;
000062                                  STATE$  OPSEP                   ; Skip separators between operands
000062                                    TRAN$   !SKIPSP               ; Skip trailing spaces
000062                                  STATE$
000062                                    TRAN$   COMMA                 ; Ensure separating comma
000062                                  STATE$
000062                                    TRAN$   !SKIPSP               ; Skip leading spaces
000062                                  STATE$
000062                                    TRAN$   $LAMDA,$EXIT          ; RETURN

000062                                  STATE$  SKIPSP                  ; Skip leading spaces and tabs
000062                                    TRAN$   $BLANK,SKIPSP
000062                                    TRAN$   $LAMDA,$EXIT          ; RETURN

000062                                  STATE$  END                     ; End-of-line test
000062                                    TRAN$   $EOS,$EXIT
000062                                    TRAN$   COMENT,$EXIT
```

```
VFPRNW
Symbol

ALNET
A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
BNQST
BRA   =
BRAST
CERR  =
CFGBF =
CFGSZ =
CFLIN =
CKNET
CLRSTR
COMENT
COMMA
C$$CKP
C$$ORE
C$$RSH
DBLQT
DQUOTE
D$$BUG
D$$ISK
D$$L11
D$$YNC
D$$YNM
END
E$$XPR
FMT10
FMT8
FM.10
FM.8
F$$LVL
F.RSIZ
G$$TPP
G$$TSS
G$$TTK
G$$WRD
HF$DLM
HF$GWY

. ABS.

DATA
$STATE
$KTAB
$KSTR
Errors

*** As

Wo k
W  k
S ze c
Size c
```

```
                                                           I 5
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40  Page 23-1
Symbol table

ALNET   000540R         HF$HOS= 000004          L$$DRV= 000000          RNPASS= ****** GX       X$$DBT= 000000
A$$CHK= 000000          HF$XDF= 000020          L$$P11= 000001          RNPL  = ****** GX       $ALPHA= 000022
A$$CPS= 000000          H$CUG   000010          L$$11R= 000000          RNTL  = ****** GX       $ALPOL= ****** GX
A$$PRI= 000000          H$DST   000012          MAXID = 000020          RNUL  = ****** GX       $ANY  = 000020
A$$TRP= 000000          H$D29   000014          MAXSTR= 000020          RNUSER= ****** GX       $BFR  = ****** GX
BNQST   000226R     003 H$FLG   000000          MOVE  = ****** GX       RNWDF   000000R     003 $BLANK= 000006
BRA   = 000074          H$GLEN  000104          MXACAC= 000020          RNWEND  000354RG        $CEACC= ****** GX
BRAST   000220R     003 H$GLT   000044          MXACPA= 000010          RNWKW   000000RG    004 $DIGIT= 000024
CERR  = ****** GX       H$GNAM  000050          MXACUS= 000020          RNWNET  000062R         $DNUMB= 000014
CFGBF = ****** GX       H$GNML= 000020          MXNOID= 000006          RNWNOD  000144R         $EOS  = 000012
CFGSZ = ****** GX       H$GPT   000046          M$$CRB= 000124          RNWST   000000RG    003 $ERRRA  000000R     002
CFLIN = ****** GX       H$HITS  000034          M$$CRX= 000000          RTSPC = ****** GX       $ERRRB  000026R     002
CKNET   000406R         H$HLEN  000044          M$$FCS= 000000          R$ACC   000064          $ERRRC  000050R     002
CLRSTR  001012R         H$LBDA  000070          M$$MGE= 000000          R$ACL   000063          $ERRRX  000104R     002
COMENT= 000073          H$LBDN  000072          M$$NET= 000000          R$NML   000002          $ERR1T= ****** GX
COMMA = 000054          H$LDTE  000002          M$$OVR= 000000          R$NOD   000023          $EXIT = 000000
C$$CKP= 000000          H$LEN   000042          NQST    000300R     003 R$NWK   000003          $FAIL = 177777
C$$ORE= 000400          H$LOTS  000032          N$$ACC= 000001          R$NWL   000104          $GETAD= ****** GX
C$$RSH= 177564          H$NETW  000024          N$$BUF= 000001          R$NXT   000000          $GETRV= ****** GX
DBLQT   000266R     003 H$NML = 000006          N$$LDV= 000001          R$PAL   000052          $LAMDA= 000000
DQUOTE= 000042          H$NPT   000022          N$$MCP= 000001          R$PAS   000053          $NUMBR= 000002
D$$BUG= 177514          H$PTB   000020          N$$MLL= 000001          R$USL   000031          $PSIPT= ****** GX
D$$ISK= 000000          H$PVC   000006          N$$MOV= 000010          R$USR   000032          $PUTAD= ****** GX
D$$L11= 000001          H$RDTE  000004          N$$NCT= 000001          R$$DER= 000000          $QRNA   000006RG
D$$YNC= 000000          H$RNW   000042          N$$PEM= 000001          R$$K11= 000001          $QRNW   000000RG
D$$YNM= 000000          H$SVC   000036          OPSEP   000140R     003 R$$SND= 000000          $RADDR= ****** GX
END     000166R     003 H$TRB   000016          P$$P45= 000000          R$$11M= 000000          $RAD50= 000016
E$$XPR= 000000          H$XAVL  000100          P$$WRD= 000000          SKIPSP  000156R     003 $RBLCK= ****** GX
FMT10 = ****** GX       H$XBIA  000074          QBRAST  000244R     003 SPFILL= ****** GX       $SAVAL= ****** GX
FMT8  = ****** GX       H$X29C  000040          QST     000254R     003 STORCH  001020R         $STRNG= 000004
FM.10 = 000000          I$$RAR= 000000          Q$$OPT= 000010          STRBUF  000165R     002 $SUBXP= 000010
FM.8  = 000000          I$$RDN= 000000          RNAACT  000312R         STRING  000176R     003 $$$FLG= 177777
F$$LVL= 000001          KET   = 000076          RNACCT= ****** GX       STRLEN  000164R     002 $$$KEY= 000001
F.RSIZ= ****** GX       K$$CNT= 177546          RNADF   000040R     003 SYNERR= ****** GX       $$$R  = 000010
G$$TPP= 000000          K$$CSR= 177546          RNAL  = ****** GX       SYSFDB= ****** GX       $$$STA= 000000
G$$TSS= 000000          K$$LDC= 000000          RNAPAS  000250R         S$$WRG= 000000          $$$TMP= 000007R     005
G$$TTK= 000000          K$$TPS= 000074          RNAUSR  000206R         S$$YSZ= 007600          .PCHAR= ****** GX
G$$WRD= 000000          LD$LP = 000000          RNETWK= ****** GX       T$$KMG= 000000          .PSTCN= ****** GX
HF$DLM= 000002          LNKEND= ****** GX       RNNL  = ****** GX       T$$MIN= 000000          .PSTPT= ****** GX
HF$GWY= 000010          L$$ASG= 000000          RNNODE= ****** GX       V$$CTR= 001000          .TPARS= ****** GX


. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        001050    001   (RW,I,LCL,REL,CON)
DATA    000206    002   (RW,D,LCL,REL,CON)
$STATE  000320    003   (RW,D,LCL,REL,CON)
$KTAB   000004    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13802 Words  ( 54 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
```

```
                                                .SBTTL  X29$DF ACTION ROUTINES

                                        ;
                                        ; MAXIMUM CIRCUITS
                                        ;
000052  005767  000000G                 X29MXC: TST     .PNUMH          ; SINGLE WORD VALUE?
000056  001007                                  BNE     101$            ; NO- ILLEGAL
000060  105767  000001G                         TSTB    .PNUMB+1        ; SINGLE BYTE VALUE?
000064  001004                                  BNE     101$            ; NO- ILLEGAL
000066  116767  000000G 000206'                 MOVB    .PNUMB,MAXVC    ; SAVE MAXIMUM CIRCUITS VALUE
000074                                          RETURN                  ; RETURN

000076                                  101$:   EMSG$R  9B              ; ILLEGAL MAX CIRCUITS VALUE

                                        ;
                                        ; COUNTER TIMER
                                        ;
000104  005767  000000G                 X29CTM: TST     .PNUMH          ; SINGLE WORD VALUE?
000110  001004                                  BNE     101$            ; NO- ILLEGAL
000112  016767  000000G 000210'                 MOV     .PNUMB,CTIMR    ; SAVE COUNTER TIMER VALUE
000120                                          RETURN                  ; RETURN

000122                                  101$:   EMSG$R  9C              ; ILLEGAL COUNTER TIMER VALUE

                                        ;
                                        ; END OF X29$DF MACRO
                                        ;
000130  016700  000000G                 X29END: MOV     $PSIPT,R0       ; POINT TO HOME BLOCK
000134  001467                                  BEQ     101$            ; THIS, OF COURSE, CAN NEVER HAPPEN
000136  062700  000040                          ADD     #H$X29C,R0      ; POINT TO X29 LISTHEAD
000142                                          GETRV   R0,#2           ; READ IN LISTHEAD
000160  005767  000000G                         TST     $BFR            ; X.29 DATABASE ALREADY EXIST?
000164  001053                                  BNE     101$            ; YES- ILLEGAL 2ND OCCURRENCE

000166  012701  000016                          MOV     #Q$LEN,R1       ; GET LENGTH OF X.29 DATABASE
000172                                          CALL    $ALL16          ; ALLOCATE FROM DSR
000176  103451                                  BCS     102$            ; BR IF UNABLE TO ALLOCATE
000200  012702  000000G                         MOV     #$BFR,R2        ; POINT TO X29 BLOCK
000204  012701  000007                          MOV     #Q$LEN/2,R1     ; GET LENGTH IN WORDS
000210  005022                          10$:    CLR     (R2)+           ; ZERO THE BLOCK
000212  005301                                  DEC     R1              ; MORE TO ZERO?
000214  001375                                  BNE     10$             ; YES, KEEP GOING

000216  016767  000206' 000002G                 MOV     MAXVC,Q$MXVC+$BFR ; STORE MAXIMUM CIRCUITS VALUE
000224  016767  000210' 000000G                 MOV     CTIMR,Q$CTIM+$BFR ; STORE COUNTER TIMER VALUE
000232  005067  000010G                         CLR     Q$TCLZ+$BFR     ; STORE TIME COUNTERS LAST ZEROED
000236                                          PUTRC   R0,#Q$LEN       ; WRITE UPDATED X29 BLOCK
000254  016701  000000G                         MOV     $PSIPT,R1       ; GET HOME BLOCK POINTER
000260  062701  000040                          ADD     #H$X29C,R1      ; POINT TO X29 LISTHEAD
000264                                          GETRV   R1,#2           ; READ IN X29 LISTHEAD
000302  010067  000000G                         MOV     R0,$BFR         ; STORE POINTER IN HOME BLOCK
000306                                          PUTRC                   ; REWRITE UPDATED LISTHEAD
000312                                          RETURN                  ; RETURN

000314                                  101$:   EMSG$R  9A              ; ILLEGAL OCCURRENCE OF X29$DF
000322                                  102$:   EMSG$R  9D              ; X.29 BLOCK ALLOCATION FAILURE

```

```
    144                                              .SBTTL  TPARS STATE TABLES
    145                                     ;
    146                                     ; TPARS STATE TABLES
    147                                     ;
    148 000060                                       ISTAT$  X3PST,X3PKW
    149
    150                                     ;
    151                                     ; X3P$DF
    152                                     ;
    153 000060                                       STATE$  X3PDF
    154 000060                                        TRAN$   %X3P$DF%,,,1,SYNERR
    155
    156 000060                                       STATE$
    157 000060                                        TRAN$   $NUMBR,,BLKDEF          ; DEFAULT BLOCK SIZE
    158
    159 000060                                       STATE$
    160 000060                                        TRAN$   <',>
    161
    162 000060                                       STATE$
    163 000060                                        TRAN$   $NUMBR,,BLKMAX          ; MAXIMUM BLOCK SIZE
    164
    165 000060                                       STATE$
    166 000060                                        TRAN$   <',>
    167
    168 000060                                       STATE$
    169 000060                                        TRAN$   $NUMBR,,WNDDEF          ; DEFAULT WINDOW SIZE
    170
    171 000060                                       STATE$
    172 000060                                        TRAN$   <',>
    173
    174 000060                                       STATE$
    175 000060                                        TRAN$  $NUMBR,$EXIT,WNDMAX      ; MAXIMUM WINDOW SIZE
```

```
VHDAT - HELP SUBROUTINE DATA    MACRO V05.03b Saturday 29-Jun-85 02:42  Page 5-1
Symbol table

A$$CHK= 000000          FILL   = 000012         M$$CRB= 000124          Q$$OPT= 000010          T$$MIN= 000000
A$$CPS= 000000          F$$LVL= 000001          M$$CRX= 000000          R$$DER= 000000          UICA     000177R
A$$PRI= 000000          G$$TPP= 000000          M$$FCS= 000000          R$$K11= 000001          UICL   = 000005
A$$TRP= 000000          G$$TSS= 000000          M$$MGE= 000000          R$$SND= 000000          V$$CTR= 001000
C$$CKP= 000000          G$$TTK= 000000          M$$NET= 000000          R$$11M= 000000          X$$DBT= 000000
C$$ORE= 000400          G$$WRD= 000000          M$$OVR= 000000          S$$WRG= 000000          $HDFNB   000156RG
C$$RSH= 177564          I$$RAR= 000000          N      = 000003         S$$YSZ= 007600          $HDSP    000150RG
DATSET  000160R         I$$RDN= 000000          N$$ACC= 000001          S.ANM1= 000014          $HEFN  = 000001 G
DEVA    000174R         K$$CNT= 177546          N$$BUF= 000001          S.ANM2= 000005          $HFDB    000004RG
DEVL  = 000003          K$$CSR= 177546          N$$LDV= 000001          S.FDB  = 000140         $HLUN1= 000004 G
D$$BUG= 177514          K$$LDC= 000000          N$$MCP= 000001          S.FNAM= 000006          $HLUN2= 000001 G
D$$ISK= 000000          K$$TPS= 000074          N$$MLL= 000001          S.FNB  = 000036         $HMOD    000000RG
D$$L11= 000001          LD$LP  = 000000         N$$MOV= 000010          S.FNBW= 000017          $HNAME   000152RG
D$$YNC= 000000          L$$ASG= 000000          N$$NCT= 000001          S.FNTY= 000004          $$$    = 000003
D$$YNM= 000000          L$$DRV= 000000          N$$PEM= 000001          S.FTYP= 000002          ...PC1= 000010R
E$$XPR= 000000          L$$P11= 000001          P$$P45= 000000          S.NFEN= 000020          ...TPC= 000140
FILA    000204R         L$$11R= 000000          P$$WRD= 000000          T$$KMG= 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000216    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10536 Words  ( 42 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.14
SY:VHDAT.V2,[132,134]VHDAT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VHDAT
```

```
VHELP

SYMBO

SYMBO

OP$ST
OP$TP
OP$TR
OP$UC
OP$US
OP$VE
OP$VE
OP$XA
OP$XP
OP$X5
OP$X9
QF$AC
QF$AL
QF$KN
QF$LO
RQB
TY$CH
TY$EV
TY$ST
TY$SU
WC.CN
WC.EV
WC.TR
WC.UN
$HELP
$VHEL
```

```
VHELP      CREATED BY  MACRO  ON 29-JUN-85 AT 02:42       PAGE 4
SYMBOL CROSS REFERENCE                                     CREF    04.00

SYMBOL   VALUE              REFERENCES

OP$STA   = 000004           #5-68
OP$TPA   = 000020           #5-68
OP$TRI   = 000040           #5-68
OP$UCS   = 000020           #5-68
OP$USE   = 000002           #5-68
OP$VEC   = 000040           #5-68
OP$VER   = 000040           #5-68
OP$XAC   = 000001           #5-68
OP$XPR   = 000002           #5-68
OP$X5S   = 000004           #5-68
OP$X9S   = 000010           #5-68
QF$ACT   = 000001           #5-68
QF$ALL   = 000002           #5-68
QF$KNO   = 000004           #5-68
QF$LOO   = 000010           #5-68
RQB      = ******   GX       6-84
TY$CHA   = 000400           #5-68
TY$EVE   = 001000           #5-68
TY$STA   = 002000           #5-68
TY$SUM   = 000000           #5-68
WC.CNT   = 000400           #5-68
WC.EVE   = 004000           #5-68
WC.TRI   = 002000           #5-68
WC.UNT   = 001000           #5-68
$HELP    = ******   GX       6-85
$VHELP     000000 RG        #6-83
```

```
VIMI
SYMB
SYMB
$VIM
$$


$$$
$$$$
.CEA
.MGM
.PDV
```

```
VIMI       CREATED BY  MACRO  ON 29-JUN-85 AT 02:42     PAGE 2
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE                REFERENCES

$VIMI      000000 RG          #5-99
$$       = 177777              5-321     5-321    #5-321     5-322     5-322    #5-322     5-323     5-323    #5-323
                              #5-324     5-325     5-325    #5-325    #5-326     5-327     5-327    #5-327     5-328
                               5-328    #5-328     5-329     5-329    #5-329     5-330     5-330    #5-330     5-331
                               5-331    #5-331    #5-332
$$$      = 001346 R           #5-321     5-324    #5-325     5-326    #5-327     5-332
$$$$     = 002004 R           #5-324    #5-326    #5-332
.CEAVL   = ******   GX         5-282
.MGMGE   = ******   GX         5-118     5-228
.PDVTA   = ******   GX         5-103
```

VINI
LOCAL

J 10

```
     54                                         .SBTTL  LOCAL DATA
     55
     56 000000     114     102     060  DEFDEV: .ASCIZ  /LB0:[/
        000003     072     133     000
     57                                         .EVEN
     58
```

```
    626 003130  016767  000000G 000000G          MOV     .CRAVL,$RADDR    ; SET ADDRESS OF RSX POOL
    627 003136  062767  177776  000000G          ADD     #-2,$RADDR       ; ADJUST TO POINT AT ROUNDING FACTOR
    628 003144                                   GETRV   ,#6              ; READ IN 3 WORD LISTHEAD
    629
    630                                          .IF DF  R$$MPL                                                  ;[MP01]
    631                                          CMP     #-1,..POLHD      ; SECONDARY POOL SUPPORTED             ;[MP01]
    632                                          BEQ     14$              ; IF NO - BRANCH                       ;[MP01]
    633                                          SWTCHI  #$POLHD          ; POINT TO CORRECT INTERNAL CELL       ;[MP01]
    634                                          GETRV   .POLHD           ; READ LISTHEAD                        ;[MP01]
    635                                  14$:    CMP     #-1,..ICAVL      ; ICB AVAILABLE LIST SUPPORTED ?       ;[MP01]
    636                                          BEQ     15$              ; IF NO - BRANCH                       ;[MP01]
    637                                          SWTCHI  #$ICAVL          ; POINT TO CELL                        ;[MP01]
    638                                          GETRV   .ICAVL,#4        ; READ LISTHEAD AND SIZE               ;[MP01]
    639                                          MOV     .MPRO,R3         ; MULTIPROCESSOR SUPPORT ?             ;[MP01]
    640                                          BLT     15$              ; IF NO - BRANCH                       ;[MP01]
    641                                  16$:    DEC     R3               ; FIRST PROCESSOR ALREADY SET UP       ;[MP01]
    642                                          BEQ     15$              ; IF NO MORE - BRANCH                  ;[MP01]
    643                                          MOV     R3,R2            ; COPY CPU NUBER                       ;[MP01]
    644                                          ASL     R2               ; MULTIPLY BY 8.                       ;[MP01]
    645                                          ASL     R2               ; ...                                  ;[MP01]
    646                                          ASL     R2               ; ...                                  ;[MP01]
    647                                          MOV     $ICAVL,$ICAVL(R2) ; COPY ADDRESS                        ;[MP01]
    648                                          ADD     #2,R2            ; GET BLOCK ADDRESS                    ;[MP01]
    649                                          MOV     $ICAVL+2,$ICAVL(R2) ; COPY BLOCK ADDRESS                ;[MP01]
    650                                          BR      16$              ; CONTINUE                             ;[MP01]
    651
    652                                          .ENDC                                                           ;[MP01]
    653
    654 003156  022767  177777  000000G  15$:    CMP     #-1,..DLCOV      ; WAS DLC OVERHEAD DEFINED ?
    655 003164  001002                           BNE     30$              ; IF NE, YES
    656 003166  005067  000000G                  CLR     .DLCOV           ; ELSE, DEFAULT TO ZERO
    657 003172  022767  177777  000000G  30$:    CMP     #-1,..CRCOV      ; WAS CRC OVERHEAD DEFINED ?
    658 003200  001002                           BNE     40$              ; IF NE, YES
    659 003202  005067  000000G                  CLR     .CRCOV           ; ELSE, DEFAULT TO ZERO
    660
    661 003206                           40$:    SWTCHI                   ; SWITCH BACK TO DEFAULT BUFFER
    662
    663 003214                                   RETURN
    664
    665                                  ;
    666                                  ; ERRORS
    667                                  ;
    668
    669 003216                                   ERMSG$  <Illegal executive size ( %O ).>
    670 003254                                   ERBLK$  ,<$BFR>
    671 003262                           110$:   ERRPT$  ,$EROUT,FATAL
```

```
VINI       CREATED BY  MACRO  ON 29-JUN-85 AT 02:43      PAGE 2
SYMBOL  CROSS REFERENCE                                  CREF    04.00

SYMBOL   VALUE              REFERENCES

FRKHD       003442 R          12-829    #12-832
F.EFBK    = 000010            11-608    *11-610
F.ERR     = 000052             8-266      9-460      9-463
F.FACC    = 000043            *8-168
F.FFBY    = 000014           *11-602
F.HIBK    = 000004            11-604     11-606
F.NRBD    = 000024            *8-220
F.RACC    = 000016            *8-177
F.RSIZ    = 000002            *8-179     *9-397     *9-439    *11-586    *11-612    *11-628
F.RTYP    = 000000             8-173     *8-176
F.URBD    = 000020            *8-177    *11-625    *11-661
F3.11S    = 000400             9-440
GTDUIC      004760 R           9-356    #14-1020
HEADR       003454 R          12-832    #12-835
L$BBLK      000360             8-179      8-180
L$BXFR      000350             8-182      8-184     *8-218
MGEXT       003644 R          12-868    #12-871
MGMGE       003656 R          12-871    #12-874
NETPF       004022 R          12-901    #12-904
NS0         003524 R          12-844    #12-847
NS1         003536 R          12-847    #12-850
NS2         003550 R          12-850    #12-853
NS3         003562 R          12-853    #12-856
NS4         003574 R          12-856    #12-859
NS5         003606 R          12-859    #12-862
NS6         003620 R          12-862    #12-865
NS7         003632 R          12-865    #12-868
NTUIC       004034 R           9-394     12-904    #12-907
OKLIST      002634 R          10-485    #10-537
PARHD       003372 R          12-817    #12-820
PATT        003764 R          12-892    #12-895
PDVTA       004046 R         #12-916
PHDR        003776 R          12-895    #12-898
PLGTH       003500 R          12-838    #12-841
PPRO        004010 R          12-898    #12-901
P.BLKS      000016            13-962
P.NAM       000004            13-952     13-954
P.REL       000014            13-960
R$$MPL    = ******             7-95       8-189      8-200      8-208      8-222      9-335      9-368      9-422     10-543
                              11-589     11-630     12-700     12-913
R.FIX     = ******   GX        8-176
SB.DEF      000002            10-482    #10-564
SB.LNK      000000            10-496    #10-563
SB.SYB      000004             9-451      9-452     10-489     10-491     10-511     10-512    #10-565
SB.VAL      000010             9-394    *10-493     10-495    #10-566
STKDP       003466 R          12-835    #12-838
SYMOV       002264 R           9-436    #10-478
SYSFDB    = ******   GX        8-167      8-173     *8-179     *8-220      8-266     *9-397     *9-439    *11-586    *11-602
                              11-604     11-606     11-608    *11-610    *11-612    *11-625    *11-628    *11-661
SYSIZ       003404 R          12-820    #12-823
SYSNM       003416 R          12-823    #12-826
SYSPMT      001314 R           8-142     #8-279
```

Table of contents

```
     9- 1324     PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
    11- 1378     GETLIN - GET SLT AND PDV ADDRESS FOR LINE
    12- 1412     CHKLOG - CHECK FOR LOGGING FUNCTION
    13- 1437     STORID - Store id-string according to system type           ;[TD001]
    14- 1500     MOVE - Move bytes                                            ;[TD001]
    15- 1522     SPFILL - Fill area with spaces                               ;[TD001]
```

```
002066  000416                             BR      5$

002070  032767  000002  000000G $IFXPR::BIT     #OP$XPR,$OPTON   ; Ck for X25-PROTOCOL bit.
002076  001422                             BEQ     99$
002100  000411                             BR      5$

002102  032767  000004  000000G $IFX5S::BIT     #OP$X5S,$OPTON   ; Ck for X25-SERVER bit.
002110  001415                             BEQ     99$
002112  000404                             BR      5$

002114  032767  000010  000000G $IFX9S::BIT     #OP$X9S,$OPTON   ; Ck for X29-SERVER bit.
002122  001410                             BEQ     99$
002124  012700  002162'         5$:        MOV     #IFMODT,R0       ; MODULE function codes.

002130                          10$::      WHILE next table entry <> 0
002130  112001                               MOVB  (R0)+,R1
002132  001404                               BEQ   19$
                                ;          DO
002134                          20$::        IF this entry matches the current function code, THEN succeed.
002134  120167  000000G                        CMPB R1,RQBST
002140  001403                                 BEQ 100$
002142                          29$::        ENDIF

002142  000772                             BR      10$
002144                          19$::      ENDWHILE

002144  062716  000002          99$:       ADD     #2,(SP)          ; No match, so fail.
002150                          100$:      RETURN

002152     016     017     030  IFALIT: .BYTE   LS.ALI, LX.ALI, LD.ALI, 0
002155     000
002156     020     021     031  IFLOGT: .BYTE   LS.LOG, LX.LOG, LD.LOG, 0
002161     000
002162     036     037     000  IFMODT: .BYTE   LS.MOD, LD.MOD, 0
                                        .EVEN
                                        .DSABL  LSB
                                ;
                                ; Line name
                                ;
002166  012702  000003          $LNAME::MOV     #3,R2            ; Set max field length
002172                                  CALL    RAD              ; Convert up to 3 characters
002176  103411                          BCS     101$             ; If cs, more than 3
002200                                  CALL    CHKLOG           ; Check for logging function
002204  103003                          BCC     10$              ; Br if logging function
002206  010067  000002G                 MOV     R0,RQB+LR.LIN    ; Set line name
002212  000402                          BR      20$              ; And exit
002214  010067  000030G         10$:    MOV     R0,RQB+LO.LID    ; Set line name for logging
002220                          20$:    RETURN
002222  062716  000002          101$:   ADD     #2,(SP)
002226                                  RETURN

                                ;
                                ; Listen timer
                                ;
002230                          $LTM::
002230  005767  000000G                 TST     .PNUMH           ; Must be a word value
002234  001004                          BNE     101$             ; If ne, illegal
```

```
                                     ;
                                     ; Get up to 6 RAD50 characters
                                     ;
006052  016700  000000G              RAD:    MOV     .PSTPT,R0       ; Get field address
006056  016701  000000G                      MOV     .PSTCN,R1       ; And length
006062  160102                               SUB     R1,R2           ; Max characters or less?
006064  100427                               BMI     40$             ; If mi, no (c-bit is set)
006066  010246                               MOV     R2,-(SP)        ; Save fill length
006070  012702  000064'                      MOV     #ASCII,R2       ; Get ascii workarea address
006074  112022                       10$:    MOVB    (R0)+,(R2)+     ; Copy the field
006076  005301                               DEC     R1              ;       ...
006100  001375                               BNE     10$             ;       ...
006102  012601                               MOV     (SP)+,R1        ; Blank fill needed?
006104  001404                               BEQ     30$             ; If eq, no
006106  112722  000040               20$:    MOVB    #' ,(R2)+       ; Fill with blanks
006112  005301                               DEC     R1              ;       ...
006114  001374                               BNE     20$             ;       ...
006116  012700  000064'              30$:    MOV     #ASCII,R0       ; Convert to rad50
006122  010701                               MOV     PC,R1           ;       ...
006124                                       CALL    $CAT5           ;       ...
006130  010146                               MOV     R1,-(SP)        ; Save first half
006132  010701                               MOV     PC,R1           ; Convert second half
006134                                       CALL    $CAT5           ;       ...
006140  012600                               MOV     (SP)+,R0        ; Restore first half
006142  000241                               CLC                     ; Indicate success
006144                               40$:    RETURN

                                     ;
                                     ; Find a DCB address
                                     ;
006146                               FNDDCB:
006146  016767  000000G 000000G      10$:    MOV     $BFR,$RADDR     ; Get next DCB in list
006154  000261                               SEC                     ; Assume no more
006156  001411                               BEQ     20$             ; If eq, end of list
006160                                       GETRV   ,#D.PCB         ; Read in DCB
006172  020067  000000C                      CMP     R0,$BFR+D.NAM   ; Is this the right DCB ?
006176  001363                               BNE     10$             ; If ne, no
006200  000241                               CLC                     ; Else, indicate success
006202                               20$:    RETURN
```

```
                                        .SBTTL  ALPVC1 - Allocate PVC name block [old format] & circuit block    ;[TD001]
                                ;+                                                                               ;[TD001]
                                ; *** ALPVC1 - Allocate PVC name block [old format] & circuit block              ;[TD001]
                                ;                                                                                ;[TD001]
                                ; INPUTS                                                                         ;[TD001]
                                ;       R1 - Port number allocated to PVC                                        ;[TD001]
                                ;       All parsed fields for PVC                                                ;[TD001]
                                ;                                                                                ;[TD001]
                                ; OUTPUTS                                                                        ;[TD001]
                                ;       Registers preserved                                                      ;[TD001]
                                ;       Carry clear PVC block linked into chain                                  ;[TD001]
                                ;       Carry set if unable to allocate PVC or X25 circuit block                 ;[TD001]
                                ;-                                                                               ;[TD001]
 001344                         ALPVC1:                                                                          ;[TD001]
 001344                                 CALL    $SAVAL                  ; Save registers                         ;[TD001]
 001350                                 SAVMAP                          ; Save current mapping                   ;[TD001]
 001354  010146                         MOV     R1,-(SP)                ; Save port number                       ;[TD001]
 001356  012701  000016                 MOV     #C$LEN,R1               ; Get length of PVC name block           ;[TD001]
 001362                                 CALL    $ALPOL                  ; Attempt to allocate block              ;[TD001]
 001366  012603                         MOV     (SP)+,R3                ; Get port number back                   ;[TD001]
 001370  103451                         BCS     50$                     ; Branch if allocation failed            ;[TD001]
 001372  010005                         MOV     R0,R5                   ; Save unmapped address of allocation    ;[TD001]
 001374                                 CEACC$  R0,R4                   ; Map to allocation                      ;[TD001]
 001404  012702  000002G                MOV     #$BFR+C$NAM,R2          ; Target for PVC name                    ;[TD001]
 001410  012701  000006                 MOV     #PVNAMX,R1              ; Length of PVC name                     ;[TD001]
 001414  012700  000027'                MOV     #PVCNAM,R0              ; Source of PVC name                     ;[TD001]
 001420                                 CALL    MOVE                    ; Move PVC name                          ;[TD001]
 001424  116767  000025' 000015G        MOVB    PVCFLG,$BFR+C$FLG       ; Move PVC flags                         ;[TD001]
 001432  016767  000004' 000010G        MOV     PVCLCN,$BFR+C$LCN       ; Move channel number                    ;[TD001]
 001440  016767  000000G 000012G        MOV     DTEDES,$BFR+C$DTE       ; Move DTE descriptor                    ;[TD001]
 001446  110367  000014G                MOVB    R3,$BFR+C$PORT          ; Move port number                       ;[TD001]
 001452                                 PUTAD   R4,#C$LEN               ; Write out PVC name block               ;[TD001]
                                ;                                                                                ;[TD001]
                                ; Attempt to allocate X25 circuit block                                          ;[TD001]
                                ;                                                                                ;[TD001]
 001470                                 CALL    ALLXCB                  ; Allocate block                         ;[TD001]
 001474  103407                         BCS     50$                     ; Exit if no store                       ;[TD001]
                                ;                                                                                ;[TD001]
                                ; Link PVC into chain                                                            ;[TD001]
                                ;                                                                                ;[TD001]
 001476  016700  000000G                MOV     $PSIPT,R0               ; Form address of listhead               ;[TD001]
 001502  062700  000006                 ADD     #H$PVC,R0                                                        ;[TD001]
 001506                                 CALL    LNKEND                  ; Link PVC name (address of PVC in R5)   ;[TD001]
                                ;                                                                                ;[TD001]
 001512  000241                         CLC                             ; Signal success                         ;[TD001]
 001514                         50$:    RESMAP                          ; Restore mapping                        ;[TD001]
 001520                                 RETURN                                                                   ;[TD001]
```

```
VFPRDT - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:39
Table of contents
```

```
     5-   55     MACRO DEFINITIONS
     6-   78     LOCAL SYMBOL DEFINITIONS
     7-   86     LOCAL DATA
     8-  100     ERROR MESSAGES
     9-  113     LOOK FOR RDT$DF MACRO
    10-  142     TPARS STATE TABLES
    11-  174     RDT$DF ACTION ROUTINES
    13-  274     CHKRDT - CHECK FOR UNIQUE RDT
```

```
VFPRDT

SYMBOL

SYMBOL

$EXIT
$FAIL
$GETAD
$GETRV
$GPRM
$LAMDA
$NUMBR
$PSIPT
$PUTAD
$QRDT
$RADDR
$RAD50
$RBLCK
$RONLY
$SAVAL
$SAVRG
$STRNG
$SUBXP
$$$FLG
$$$KEY
.PSTCN
.PSTPT
.TPARS
```

```
VFPRDT      CREATED BY  MACRO  ON 29-JUN-85 AT 02:39     PAGE 2

SYMBOL CROSS REFERENCE                                   CREF    04.00

SYMBOL  VALUE                REFERENCES

$EXIT   = 000000             #10-146
$FAIL   = 177777             #10-146
$GETAD  = ******    GX        13-298
$GETRV  = ******    GX        13-291
$GPRM   = ******              10-146
$LAMDA  = 000000             #10-146
$NUMBR  = 000002             #10-146
$PSIPT  = ******    GX        12-262        13-289
$PUTAD  = ******    GX        12-261
$QRDT     000000 RG          #9-125
$RADDR  = ******    GX       *12-261      *13-291      *13-298
$RAD50  = 000016             #10-146
$RBLCK  = ******    GX        12-239      *12-265       13-293      *13-308
$RONLY  = ******              10-146       10-146       10-146
$SAVAL  = ******    GX        12-236       13-292
$SAVRG  = ******    GX        12-227
$STRNG  = 000004             #10-146
$SUBXP  = 000010             #10-146
$$$FLG  = 177777             #10-146
$$$KEY  = 177777             #10-146
.PSTCN  = ******    GX        11-179
.PSTPT  = ******    GX        11-188
.TPARS  = ******    GX        9-131
```

```
                                ;
                                ; Parse a string
                                ;
                                ; String can be sequence of alphanumeric characters terminated by non alphan.
                                ;             or sequence of characters in quotes ("...")
                                ;             or sequence of characters in angle brackets (<...>)
                                ;             or sequence of characters in quotes and angle brackets (<"...">)
                                ;
       000062                           STATE$  STRING                  ; Parse quoted/unquoted string
       000062                             TRAN$   BRA,BRAST,CLRSTR      ; Check for bracketed string
       000062                             TRAN$   DQUOTE,QST,CLRSTR     ; Check for quoted string
       000062                             TRAN$   $LAMDA,NQST,CLRSTR    ; Must be unbracketed, unquoted

       000062                           STATE$  BRAST                   ; String in brackets
       000062                             TRAN$   DQUOTE,QBRAST         ; Check for quoted, bracketed
       000062                             TRAN$   $LAMDA
       000062                           STATE$  BNQST                   ; Bracketed unquoted string
       000062                             TRAN$   KET,$EXIT             ; Exit on end bracket
       000062                             TRAN$   $EOS,$EXIT            ; Exit at end-of-string
       000062                             TRAN$   $ANY,BNQST,STORCH     ; Accept any other character
       000062                           STATE$  QBRAST                  ; Quoted, bracketed string
       000062                             TRAN$   !QST                  ; Get quoted string
       000062                           STATE$
       000062                             TRAN$   KET,$EXIT             ; Ensure closing bracket

       000062                           STATE$  QST                     ; Quoted string
       000062                             TRAN$   DQUOTE,DBLQT          ; If quote, check for double quote
       000062                             TRAN$   $ANY,QST,STORCH       ; Accept any other character
       000062                           STATE$  DBLQT
       000062                             TRAN$   DQUOTE,QST,STORCH     ; Recognise two quotes as quote char
       000062                             TRAN$   $LAMDA,$EXIT          ; Exit on single quote

       000062                           STATE$  NQST                    ; Unquoted string
       000062                             TRAN$   $ALPHA,NQST,STORCH    ; Accept alpha-numerics only
       000062                             TRAN$   $DIGIT,NQST,STORCH
       000062                             TRAN$   $LAMDA,$EXIT          ; Exit on anything else

       000062                           STATE$                          ; End of state tables
```

```
VFPRNW
Symbol

Operat

Elapse
SY:VFP
```

```
                                                    J 5
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40  Page 23-2
Symbol table

Operating  system: RSX-11M/PLUS

Elapsed time: 00:01:02.09
SY:VFPRNW.V2,[132,134]VFPRNW/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPRNW
```

```
    208         000001                          .END
```

```
                        .SBTTL  CHN$DF STATE TABLE

                ;
                ; CHN$DF STATE TABLE
                ;
000060                  STATE$  CHNDF
000060                   TRAN$   %CHN$DF%,,,1,SYNERR

000060                  STATE$
000060                   TRAN$   $NUMBR,,CHLO            ; LOW END OF CHANNEL RANGE

000060                  STATE$
000060                   TRAN$   :END,$EXIT,CHEND
000060                   TRAN$   <',>

000060                  STATE$
000060                   TRAN$   $NUMBR,END,CHHI         ; HIGH END OF CHANNEL RANGE

                ;
                ; CHECK FOR END OF SOURCE LINE
                ;
000060                  STATE$  END
000060                   TRAN$   <';>,$EXIT
000060                   TRAN$   $EOS,$EXIT

000060                  STATE$
```

```
VHDAT      CREATED BY  MACRO  ON 29-JUN-85 AT 02:42      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

DATSET     000160 R          5-68      #5-73
DEVA       000174 R          5-73      #5-75       5-76
DEVL     = 000003            5-73      #5-76
FILA       000204 R          5-73      #5-81       5-82
FILL     = 000012            5-73      #5-82
N        = 000003           #5-68       5-68
S.FDB    = 000140            5-68       5-68
UICA       000177 R          5-73      #5-78       5-79
UICL     = 000005            5-73      #5-79
$HDFNB     000156 RG        #5-68
$HDSP      000150 RG        #5-68
$HEFN    = 000001  G        #5-68
$HFDB      000004 RG        #5-68
$HLUN1   = 000004  G        #5-68
$HLUN2   = 000001  G        #5-68
$HMOD      000000 RG        #5-68
$HNAME     000152 RG        #5-68
$$$      = 000003           #5-68       5-68
...PC1   = 000010 R         #5-68
```

```
VHELP      CREATED BY  MACRO  ON 29-JUN-85 AT 02:42      PAGE 5
MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME         REFERENCES

CALLR              #5-59        6-85
VNPDF$             #5-59        5-68
```

```
VIMI
MACR
MACR
CALL
ERBL
ERMS
ERRF
GETA
GETR
HWDD
PHBD
PUTR
RAND

RETU
SWTC
VNPD
```

```
VIMI        CREATED BY  MACRO  ON 29-JUN-85 AT 02:42      PAGE 3

MACRO CROSS REFERENCE                                     CREF   04.00

MACRO NAME        REFERENCES

CALL               5-100     5-103     5-127     5-185     5-244     5-284     5-295     5-315     5-316     5-317
ERBLK$            #5-82      5-324     5-326     5-332
ERMSG$            #5-82      5-321     5-322     5-323     5-325     5-327     5-328     5-329     5-330     5-331
ERRPT$            #5-82      5-315     5-316     5-317
GETAD             #5-82      5-244
GETRV             #5-82      5-103     5-127     5-185     5-284     5-295
HWDDF$            #5-82      5-84
PHBDF$            #5-92      5-93
PUTRC             #5-82
RAND              #5-103     5-103    #5-127     5-127    #5-185     5-185    #5-244     5-244    #5-284     5-284
                  #5-295     5-295
RETURN             5-306
SWTCHI            #5-82      5-101     5-105     5-172     5-224     5-281     5-288
VNPDBG            #5-95      5-99
```

VINI
CONST

```
                                        .SBTTL  CONSTANT DEFINITIONS

                                ;
                                ; CONVERSION FLAG FOR .PPASC ROUTINE
                                ;
          000001                UICCFG = 1                      ; NO ZERO SUPPRESSION, INCLUDE SEPARATORS
```

```
                                ;+
                                ;       BELOW FIND A TABLE OF SYMBOLS THAT ARE LIKELY TO BE NEEDED
                                ;       AT SOME TIME DURING THE RUNNING OF VNP. THESE SYMBOLS MUST
                                ;       BE LOOKED UP IN EACH RSX11M.STB & CEX.STB ENCOUNTERED SINCE
                                ;       THEIR VALUES CHANGE FROM SYSTEM TO SYSTEM. THE STRUCTURE BELOW
                                ;       IS USED TO LOOK UP THE SYMBOLS. AFTER LOOKUP, THEIR VALUES
                                ;       ARE MOVED TO THE ROOT SEGEMNET VRES AND ARE DEFINED GLOBALLY.
                                ;       GENERALLY, ANY SYMBOL WHOSE VALUE WAS LOOKED UP IN THE SYMBOL
                                ;       TABLE HAS A GLOBAL SYMBOL NAME THE STARTS WITH A '.'. IF
                                ;       THIS SYMBOL IS THE ADDRESS OF A DESIRED VALUE, AND IF THIS
                                ;       SYMBOL WILL BE REFERENCED OFTEN DURING THE RUNNING OF VNP,
                                ;       THE IT'S VALUE IS READ IN FROM THE EXEC IMAGE AND A SECOND
                                ;       SYMBOL NAME IS CREATED IN THE ROOT STARTING WITH A '$'.
                                ;       $PARHD AND $FMASK ARE SUCH SYMBOLS.
                                ;
                                ;       NOTE THAT THERE IS A ONE-TO-ONE CORRESPONDANCE BETWEEN THE
                                ;       SYMBOLS IN THIS TABLE AND THE CORRESPONDING SYMBOLS IN THE
                                ;       ROOT MODULE VRES. IF YOU ADD A SYMBOL TO THIS TABLE, THEN YOU
                                ;       BETTER ADD AN ENTRY TO THE TABLE STARTING AT '$SYMTB'. THE
                                ;       SYMBOLS THAT ARE ADDRESS OF WORDS IN THE EXEC THAT ARE
                                ;       FREQUENTLY USED BY VNP ARE LOOKED UP IN THE ROUTINE 'VSYM1'
                                ;       AND ARE LOADED INTO THE ROOT USING A NAME STARTING WITH '$'
                                ;       INSTEAD OF '.'.
                                ;-

 003276                         EXSTB:                          ; REFERENCE POINT

                                        .IF     NDF,R$$MPL

 003276  003310' 000000         S.MPR:  .WORD   CRAVL,0         ; USED ONLY TO FORCE ERROR IF ON RSX-11M-PLUS
 003302  075455  063320                 .RAD50  "S.MPR"
 003306  000000                         .WORD   0

                                        .IFF
                                MPRO:   .WORD   KDAS,0          ; NUMBER OF PROCESSORS OR -1 IF SINGLE PROC
                                        .RAD50  "M$$PRO"
                                        .WORD   0
                                KDAS:   .WORD   SCMOF,0         ; KERNAL DATA SPACE ENABLED OR -1 IF NOT
                                        .RAD50  "K$$DAS"
                                        .WORD   0
                                SCMOF:  .WORD   CPPAR,0         ; STARTING PHYSICAL ADDRESS OF 4K DATA SPACE
                                        .RAD50  "$SCMOF"
                                        .WORD   0
                                CPPAR:  .WORD   K6TAB,0         ; POINTER TO PCB OF CPU PARTITION
                                        .RAD50  "$CPPAR"
                                        .WORD   0
                                K6TAB:  .WORD   M100Q,0         ; TABLE OF POINTERS TO EACH PROCESSOR'S AREA
                                        .RAD50  "$K6TAB"
                                        .WORD   0
                                M100Q:  .WORD   CPURM,0         ; 100 MSEC TIMER OFFSET INTO CPU PARTITION
                                        .RAD50  "$M100Q"
                                        .WORD   0
                                CPURM:  .WORD   UMRPT,0         ; CPU URM MAPPING TABLE POINTER
                                        .RAD50  "$CPURM"
                                        .WORD   0
                                UMRPT:  .WORD   URMTB,0         ; PER CPU NETWORK POOL UMR POINTER
                                        .RAD50  "$UMRPT"
```

```
VINI      CREATED BY  MACRO  ON 29-JUN-85 AT 02:43      PAGE 3
SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL   VALUE          REFERENCES

S.MPR      003276 R     #12-702
TLGTH      003740 R      12-886    #12-889
TSKHD      003714 R      12-880    #12-883
UICCFG   = 000001       #6-65       14-1024
UMRHD      003670 R      12-874    #12-877
VCHECK     004130 R     #13-947
VCTR       003512 R      12-841    #12-844
VFILE      000202 R      7-88      #8-140      8-248      8-250
VF.INI   = ******   GX   7-106
VF.SS    = ******   GX   8-160      9-442
VSYMB      001340 R      7-90      #9-302
VSYM1      002662 R      7-92      #11-585
V1.NSY   = ******   GX   8-252
$AZFS    = ******   GX   9-362      9-429
$BFR     = ******   GX   8-180      8-182      8-184     *8-218      8-220      9-398      9-440     *9-451     *9-452
                        *10-511    *10-512     10-529     10-531     11-585     11-661     11-670    *13-947     13-949
                         13-952     13-954     13-960     13-962    *13-972     13-1007
$CBOMG   = ******   GX   9-410      9-415
$CEX     = ******   GX   9-423
$CLBUF   = ******   GX   8-141     *8-143     *8-146      9-330      9-343
$CLCSI   = ******   GX   8-161
$CLEVL   = ******   GX   8-152
$CLFDB   = ******   GX   9-460      9-463
$CLFLG   = ******   GX  *8-144     *8-151     *8-154
$CLIOS   = ******   GX   8-156
$CLPMT   = ******   GX   8-140     *8-142     *8-147
$CRAVL   = ******   GX   11-625
$DEVHD   = ******   GX  *11-622
$DFUIC   = ******   GX   9-347      9-354      9-360      9-402      9-420      14-1023
$EROUT   = ******   GX   7-116      8-249      8-251      8-253      8-254      9-449      9-450      9-453      10-516
                         10-519     11-671     13-986     13-988     13-991     13-993
$EXSIZ   = ******   GX  *11-615
$FMASK   = ******   GX   7-108     *11-613     11-617
$GENND   = ******   GX  *13-962    *13-963     13-964
$GENST   = ******   GX  *13-960     13-961     13-963
$GETRC   = ******   GX   8-179
$GETRV   = ******   GX   9-397      9-439      11-586     11-612     11-614     11-621     11-623     11-628     13-951
$LOWSZ   = ******   GX  *8-193     *8-196     *8-201
$NTUIC   = ******   GX  *9-398      9-408      9-413
$PARHD   = ******   GX  *11-624     13-947
$PUTRC   = ******   GX   8-219
$RADDR   = ******   GX  *8-179     *9-397     *9-439     *11-586    *11-612    *11-614    *11-621    *11-623    *11-626
                        *11-627    *13-949
$RSIZ    = ******   GX  *8-175
$RSIZE   = ******   GX   8-175     *13-948
$RSX     = ******   GX   9-325
$SAVAL   = ******   GX   14-1020
$STB     = ******   GX   9-361      9-428
$SYMBL   = ******   GX   9-364      9-431
$SYMND   = ******   GX   10-501     10-513
$SYMTB   = ******   GX   10-479
$SYSIZ   = ******   GX  *11-587     11-596     13-965
```

```
                                        .TITLE  VNCP - VNP GET COMMANDS NCP STYLE
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ; module description:
                                ;
                                ;       VNP - COMMAND LINE PARSER
                                ;
                                ; Distributed Systems Software Engineering
                                ;
                                ; Ident history:
                                ;
                                ; 1.00  27-feb-78
                                ;       Version 2.0 release
                                ;
                                ; 2.00  14-dec-79
                                ;       Decnet-11m/s v3.0
                                ;       Decnet-11m-plus v1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;

```

```
002236  016767  000000G 000034G          MOV     .PNUMB,RQB+LO.LTM ; Store the timer value.
002244                          10$:     RETURN
002246                          101$:    ERROR$  BDLIST            ; Illegal timer value.


                                ;
                                ; Multipoint active ratio
                                ;
002266  005767  000000G         $MACT::  TST     .PNUMH            ; Must be a word value
002272  001010                           BNE     101$              ; If ne, illegal value
002274  105767  000001G                  TSTB    .PNUMB+1          ; Must be a byte value
002300  001005                           BNE     101$              ; If ne, illegal value
002302  016767  000000G 000022G          MOV     .PNUMB,RQB+LO.MAC ; Store active ratio
002310  001401                           BEQ     101$              ; If zero, illegal value
002312                                   RETURN
002314                          101$:    ERROR$  ILLMAC            ; Illegal active ratio

                                ;
                                ; Multipoint dead ratio
                                ;
002334  005767  000000G         $MDEA::  TST     .PNUMH            ; Must be a word value
002340  001010                           BNE     101$              ; If ne, illegal value
002342  105767  000001G                  TSTB    .PNUMB+1          ; Must be a byte value
002346  001005                           BNE     101$              ; If ne, illegal value
002350  016767  000000G 000024G          MOV     .PNUMB,RQB+LO.MDE ; Store active ratio
002356  001401                           BEQ     101$              ; If zero, illegal value
002360                                   RETURN
002362                          101$:    ERROR$  ILLMDE            ; Illegal active ratio
                                ;
                                ; Incoming timer
                                ;
002402  005767  000000G         $INC::   TST     .PNUMH            ; Must be a word value
002406  001010                           BNE     101$              ; If ne, illegal value
002410  105767  000001G                  TSTB    .PNUMB+1          ; Must be a byte value
002414  001005                           BNE     101$              ; If ne, illegal value
002416  016767  000000G 000060G          MOV     .PNUMB,RQB+LO.INC ; Store incoming timer
002424  001401                           BEQ     101$              ; If zero, illegal value
002426                                   RETURN
002430                          101$:    ERROR$  ILLINC            ; Illegal incoming timer value
                                ;
                                ; Outgoing timer
                                ;
002450  005767  000000G         $OUT::   TST     .PNUMH            ; Must be a word value
002454  001010                           BNE     101$              ; If ne, illegal value
002456  105767  000001G                  TSTB    .PNUMB+1          ; Must be a byte value
002462  001005                           BNE     101$              ; If ne, illegal value
002464  016767  000000G 000056G          MOV     .PNUMB,RQB+LO.OUT ; Store outgoing timer
002472  001401                           BEQ     101$              ; If zero, illegal value
002474                                   RETURN
002476                          101$:    ERROR$  ILLOUT            ; Illegal outgoing timer value
                                ;
                                ; Inactivity timer
                                ;
002516  005767  000000G         $INA::   TST     .PNUMH            ; Must be a word value
002522  001010                           BNE     101$              ; If ne, illegal value
002524  105767  000001G                  TSTB    .PNUMB+1          ; Must be a byte value
002530  001005                           BNE     101$              ; If ne, illegal value
002532  016767  000000G 000062G          MOV     .PNUMB,RQB+LO.INA ; Store inactivity timer
```

```
                                        .SBTTL  PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
                                ;+
                                ;
                                ; PCKBCD - PACK A STRING OF DIGITS IN BCD FORMAT
                                ;
                                ; INPUTS:
                                ;       R0 - NUMBER OF BYTES IN PACKED ADDRESS
                                ;       R1 - ADDRESS OF BUFFER CONTAINING ASCII DIGITS TO PACK
                                ;       R2 - NUMBER OF DIGITS SPECIFIED
                                ;       R5 - ADDRESS OF BUFFER TO STORE PACKED DATA
                                ;
                                ; OUTPUTS:
                                ;       BUFFER SPECIFIED IN R5 CONTAINS DATA IN BCD FORMAT
                                ;       R3 DESTROYED
                                ;
                                ;-

006204  010503                  PCKBCD: MOV     R5,R3           ; GET ADDRESS OF PACKED DATA
006206  105023                  10$:    CLRB    (R3)+           ; INITIALIZE PACKED ADDRESS
006210  005300                          DEC     R0              ; MORE TO INITIALIZE?
006212  003375                          BGT     10$             ; BR IF YES
006214  005003                          CLR     R3              ; POSITION INIDCATOR (0=HIGH FOUR BITS)
006216  005305                          DEC     R5              ; UPDATE ADDRESS FOR FIRST TIME
006220  112100                  20$:    MOVB    (R1)+,R0        ; GET DIGIT TO PACK
006222  020027  000060                  CMP     R0,#60          ; IS THIS A VALID DIGIT?
006226  103422                          BLO     70$             ; BR IF NO (SETS CARRY)
006230  022700  000071                  CMP     #71,R0          ; STILL VALID?
006234  103417                          BLO     70$             ; BR IF NO
006236  162700  000060                  SUB     #60,R0          ; CONVERT TO BINARY
006242  005703                  40$:    TST     R3              ; STORE IN HIGH FOUR BITS?
006244  001007                          BNE     50$             ; BR IF NO
006246                                  ASL$    4,R0            ; MOVE TO HIGH FOUR BITS
006256  005203                          INC     R3              ; INDICATE STORE IN LOW FOUR BITS NEXT
006260  005205                          INC     R5              ; POINT TO NEXT BYTE
006262  000401                          BR      60$             ; BR TO STORE DECIMAL
006264  005303                  50$:    DEC     R3              ; INDICATE STORE IN HIGH FOUR BITS NEXT
006266  150015                  60$:    BISB    R0,(R5)         ; PACK THE NUMBER INTO THE BUFFER
006270  005302                          DEC     R2              ; MORE TO CONVERT?
006272  003352                          BGT     20$             ; BR IF YES
006274                          70$:    RETURN
```

```
                                         .SBTTL  FNDPRT - FIND FREE PORT ENTRY                        ;[TD001]
                                 ;+                                                                   ;**-28
                                 ;
                                 ; FNDPRT - FIND FREE PORT ENTRY
                                 ;
                                 ; INPUTS:
                                 ;         PORTNO - TOTAL NUMBER OF PORTS
                                 ;
                                 ; OUTPUTS:
                                 ;         CARRY CLEAR - FREE PORT FOUND
                                 ;                 R1 - PORT NUMBER ALLOCATED
                                 ;                 PVCPRT - PORT NUMBER
                                 ;                 PRTADD - MAPPED ADDRESS OF ENTRY IN PORT TABLE
                                 ;         CARRY SET - FREE PORT NOT FOUND
                                 ;
                                 ;-
001522                           FNDPRT: SAVRG   <R0,R2>          ; SAVE R0,R2
001526                                   SAVMAP                   ; SAVE CURRENT MAPPING
001532  016700  000000G                  MOV     $PSIPT,R0        ; GET ADDRESS OF PSI HOME BLOCK
001536  062700  000020                   ADD     #H$PTB,R0        ; POINT TO PORT TABLE LISTHEAD
001542                                   GETRV   R0,#2            ; READ IN LISTHEAD
001560                                   CEACC$  $BFR,R0          ; GET MAPPED ADDRESS
001572  016701  000000G                  MOV     PORTNO,R1        ; GET NUMBER OF PORTS
001576  006301                           ASL     R1               ; GET NUMBER OF BYTES IN PORT TABLE
001600                                   GETAD   R0,R1            ; READ IN PORT TABLE
001614  012702  000000G                  MOV     #$BFR,R2         ; POINT TO PORT TABLE
001620  005001                           CLR     R1               ; PORT NUMBER COUNTER
001622  005722                           TST     (R2)+            ; SKIP FIRST ENTRY (NOT AVAILABLE)
001624  005201                   10$:    INC     R1               ; INCREMENT PORT NUMBER
001626  005722                           TST     (R2)+            ; IS THIS A FREE PORT?
001630  001011                           BNE     15$              ; BR IF NO
001632  005742                           TST     -(R2)            ; POINT TO PORT ENTRY
001634  016767  000000G 000002'          MOV     $RBLCK,PRTBIA    ; SAVE BIAS OF PORT TABLE
001642  010267  000000'                  MOV     R2,PRTADD        ; SAVE MAPPED ADDRESS OF ENTRY
001646  110167  000024'                  MOVB    R1,PVCPRT        ; SAVE PORT NUMBER ALLOCATED
001652  000404                           BR      20$              ; AND EXIT
001654  020167  000000G          15$:    CMP     R1,PORTNO        ; END OF LIST?
001660  103761                           BLO     10$              ; BR IF NO
001662  000261                           SEC                      ; INDICATE NUMBER OF PORTS EXCEEDED
001664                           20$:    RESMAP                   ; RESTORE PREVIOUS MAPPING
001670                                   RESRG   <R2,R0>          ; RESTORE R0,R2
001674                                   RETURN                   ;
```

```
                .TITLE  VFPRDT - INITIAL CONFIG FILE SCAN ACTION ROUTINES
                .IDENT  /V05.00/

        ;
        ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
        ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
        ;
        ;
        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
        ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
        ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
        ; TRANSFERRED.
        ;
        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
        ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
        ; CORPORATION.
        ;
        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
        ;
        ;
        ; MODULE DESCRIPTION:
        ;
        ; VNP - CONFIG FILE ACTION ROUTINES
        ;
        ;
        ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
        ;
        ; IDENT HISTORY:
        ;
        ; 1.00  02-MAR-82
        ;       DECNET-11M/S V3.0
        ;       DECNET-11M-PLUS V1.0
        ;
        ; 3.00  16-APR-82
        ;       DECNET-11M V3.1
        ;       DECNET-11M-PLUS V1.1
        ;
        ; 4.00  07-NOV-83
        ;       DECNET-11M V4.0
        ;       DECNET-11M-PLUS V2.0
        ;
        ; 5.00  22-JUL-85
        ;       DECnet-11M/S V4.2
        ;       DECnet-11M-Plus V3.0
        ;       DECnet-Micro/RSX V1.0
        ;
        ;
```

```
VFPRDT

MACRO (

MACRO N

CALL

CEACC$
DBGTP$
EMSG$R
GETAD
GETRV
ISTAT$
MTRAN$
NTLER$
PHBDF$
PUTAD
RAND
RDTDF$
RESMAP
RETURN
SAVMAP
STATE$
TRAN$
```

```
                                                                     K  3
VFPRDT     CREATED BY  MACRO  ON 29-JUN-85 AT 02:39        PAGE 3
MACRO CROSS REFERENCE                                      CREF    04.00

MACRO NAME         REFERENCES

CALL                 9-131      11-208      11-216      12-231      12-232      12-236      12-237      12-241      12-261      12-264
                    13-291      13-292      13-297      13-298
CEACC$             #5-61        12-241      13-297
DBGTP$             #10-146     #10-164     #10-170
EMSG$R             #5-60         9-140      11-196      11-221      12-271      12-272
GETAD              #5-61        13-298
GETRV              #5-61        13-291
ISTAT$             #5-60        10-146
MTRAN$             #10-146
NTLER$             #5-60         8-104       8-105       8-106
PHBDF$             #5-60         5-63
PUTAD              #5-61        12-261
RAND               #12-261      12-261     #13-291      13-291     #13-298      13-298
RDTDF$             #5-60         5-64
RESMAP             #5-73        12-265      13-308
RETURN               9-135      11-192      11-209      11-217      12-266      13-309
SAVMAP             #5-69        12-239      13-293
STATE$             #5-60        10-150     #10-153     #10-156     #10-159     #10-162     #10-168     #10-172
TRAN$              #5-60       #10-151     #10-154     #10-157     #10-160     #10-163     #10-164     #10-169     #10-170
```

K 4

```
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40   Page 14
Syntax action routines

                                        .SBTTL  Syntax action routines

                                ;+
                                ; *** RNWNET - Check network name
                                ;
                                ; INPUTS
                                ;       .PSTPT - Points to parsed string
                                ;       .PSTCN - Number of characters in string
                                ;
                                ; OUTPUTS
                                ;       Username checked and stored
                                ;       Error condition if invalid username
                                ;-

 000062                         RNWNET:
 000062  016701  000000G                MOV     .PSTCN,R1               ; Get length of string
 000066  020127  000020                 CMP     R1,#MAXID               ; Check if network name too long
 000072  101021                         BHI     101$                    ; Branch if error
 000074  110167  000000G                MOVB    R1,RNTL                 ; Move network name length to data area
 000100  016700  000000G                MOV     .PSTPT,R0               ; Move network name to data area
 000104  012702  000000G                MOV     #RNETWK,R2
 000110                                 CALL    MOVE
 000114  105067  000000G                CLRB    RNNL                    ; Clear remaining field lengths
 000120  105067  000000G                CLRB    RNUL
 000124  105067  000000G                CLRB    RNPL
 000130  105067  000000G                CLRB    RNAL
 000134                                 RETURN

 000136                         101$:   EMSG$R  RA                      ; Illegal network name
```

```
VFPRNW

SYMBOL

SYMBOL

ALNET
BRA
CERR
CFGBF
CFGSZ
CFLIN
CKNET
CLRSTR
COMENT
COMMA
DQUOTE
FMT10
FMT8
FM.10
FM.8
F.RSIZ
H$RNW
KET
LNKEND
MAXID
MAXSTR
MOVE

MXACA
MXACP
MXACU
MXNOI
N$$VC
RNAAC
RNACC
RNADF
RNAL
RNAPA
RNAUS
RNETW
RNNL
RNNOD
RNPAS
RNPL
RNTL
RNUL
RNUSE
RNWDF
RNWEN
RNWKW
RNWNE
RNWNO
RNWST
RTSPC
R$ACC
R$ACL
R$NML
```

L 4

K 5

```
VFPRNW     CREATED BY  MACRO  ON 29-JUN-85 AT 02:40      PAGE 1

SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL   VALUE                REFERENCES

ALNET      000540 R      19-393    #21-448
BRA      = 000074       #5-55
CERR     = ******  GX    7-90       7-91       7-92       7-93
CFGBF    = ******  GX    9-129
CFGSZ    = ******  GX    9-128
CFLIN    = ******  GX    7-90       7-91       7-92       7-93
CKNET      000406 R      19-391    #20-413
CLRSTR     001012 R     #22-505
COMENT   = 000073       #5-54
COMMA    = 000054       #5-53
DQUOTE   = 000042       #5-52
FMT10    = ******  GX    7-90       7-91       7-92
FMT8     = ******  GX    7-93
FM.10    = 000000       #7-90      #7-91      #7-92
FM.8     = 000000       #7-93
F.RSIZ   = ******  GX   *20-417    *20-422    *21-488
H$RNW      000042        20-416     21-490
KET      = 000076       #5-56
LNKEND   = ******  GX    21-491
MAXID    = 000020       #5-62       14-271
MAXSTR   = 000020       #5-64       8-101      23-519
MOVE     = ******  GX    14-276     15-303     16-327     17-350     18-374     21-465     21-472     21-477     21-482
                         21-487
MXACAC   = 000020       #5-61       18-369
MXACPA   = 000010       #5-60       17-345
MXACUS   = 000020       #5-59       16-322
MXNOID   = 000006       #5-58       15-298     21-467
N$$VCT   = ******        20-421     21-455
RNAACT     000312 R     #18-367
RNACCT   = ******  GX    18-373
RNADF      000040 R      9-123
RNAL     = ******  GX   *14-280    *18-371     21-483
RNAPAS     000250 R     #17-343
RNAUSR     000206 R     #16-320
RNETWK   = ******  GX    14-275     20-427
RNNL     = ******  GX   *14-277    *15-300     21-470
RNNODE   = ******  GX    15-302
RNPASS   = ******  GX    17-349
RNPL     = ******  GX   *14-279    *17-347     21-478
RNTL     = ******  GX   *14-273     20-424     21-461
RNUL     = ******  GX   *14-278    *16-324     21-473
RNUSER   = ******  GX    16-326
RNWDF      000000 R      9-120
RNWEND     000354 RG    #19-390
RNWKW      000000 RG     9-127     #10-144
RNWNET     000062 R     #14-269
RNWNOD     000144 R     #15-296
RNWST      000000 RG    #10-144
RTSPC    = ******  GX    7-90       7-91       7-92       7-93
R$ACC      000064        21-486
R$ACL      000063       *21-485
R$NML      000002        20-423    *21-463
```

L 5

```
VFPX29 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 02:41  Page 9-2
Symbol table

A$$CHK= 000000            HF$HOS= 000004            I$$RAR= 000000            Q$CLEN= 000006            $BFR  = ****** GX
A$$CPS= 000000            HF$XDF= 000020            I$$RDN= 000000            Q$CTIM  000000            $BLANK= 000006
A$$PRI= 000000            H$CUG   000010            K$$CNT= 177546            Q$ICRE  000014            $DIGIT= 000024
A$$TRP= 000000            H$DST   000012            K$$CSR= 177546            Q$LEN   000016            $DNUMB= 000014
CERR  = ****** GX         H$D29   000014            K$$LDC= 000000            Q$MXAC  000012            $EOS  = 000012
CFGBF = ****** GX         H$FLG   000000            K$$TPS= 000074            Q$MXVC  000002            $ERR1T= ****** GX
CFGSZ = ****** GX         H$GLEN  000104            LD$LP = 000000            Q$TCLZ  000010            $ERR9A  000000R    002
CFLIN = ****** GX         H$GLT   000044            L$$ASG= 000000            Q$$OPT= 000010            $ERR9B  000036R    002
CTIMR   000210R    002    H$GNAM  000050            L$$DRV= 000000            RTSPC = ****** GX         $ERR9C  000076R    002
C$$CKP= 000000            H$GNML= 000020            L$$P11= 000001            R$$DER= 000000            $ERR9D  000132R    002
C$$ORE= 000400            H$GPT   000046            L$$11R= 000000            R$$K11= 000001            $EXIT = 000000
C$$RSH= 177564            H$HITS  000034            MAXVC   000206R    002    R$$SND= 000000            $FAIL = 177777
D$$BUG= 177514            H$HLEN  000044            M$$CRB= 000124            R$$11M= 000000            $GETRV= ****** GX
D$$ISK= 000000            H$LBDA  000070            M$$CRX= 000000            SYNERR= ****** GX         $LAMDA= 000000
D$$L11= 000001            H$LBDN  000072            M$$FCS= 000000            SYSFDB= ****** GX         $NUMBR= 000002
D$$YNC= 000000            H$LDTE  000002            M$$MGE= 000000            S$$WRG= 000000            $PSIPT= ****** GX
D$$YNM= 000000            H$LEN   000042            M$$NET= 000000            S$$YSZ= 007600            $PUTRC= ****** GX
END     000030R    003    H$LOTS  000032            M$$OVR= 000000            T$$KMG= 000000            $QX29   000000RG
E$$XPR= 000000            H$NETW  000024            N$$ACC= 000001            T$$MIN= 000000            $RADDR= ****** GX
FMT10 = ****** GX         H$NML = 000006            N$$BUF= 000001            V$$CTR= 001000            $RAD50= 000016
FMT8  = ****** GX         H$NPT   000022            N$$LDV= 000001            X$$DBT= 000000            $STRNG= 000004
FM.10 = 000000            H$PTB   000020            N$$MCP= 000001            X29CTM  000104R           $SUBXP= 000010
FM.8  = 000000            H$PVC   000006            N$$MLL= 000001            X29DF   000000R    003    $$$FLG= 177777
F$$LVL= 000001            H$RDTE  000004            N$$MOV= 000010            X29END  000130R           $$$KEY= 000000
F.RSIZ= ****** GX         H$RNW   000042            N$$NCT= 000001            X29KW   000000RG   004    $$$R  = 000010
G$$TPP= 000000            H$SVC   000036            N$$PEM= 000001            X29MXC  000052R           $$$STA= 000000
G$$TSS= 000000            H$TRB   000016            P$$P45= 000000            X29ST   000000RG   003    $$$TMP= 000000R    005
G$$TTK= 000000            H$XAVL  000100            P$$WRD= 000000            $ALL16= ****** GX         .PNUMB= ****** GX
G$$WRD= 000000            H$XBIA  000074            Q$ACTC  000004            $ALPHA= 000022            .PNUMH= ****** GX
HF$DLM= 000002            H$X29C  000040            Q$AUC   000006            $ANY  = 000020            .TPARS= ****** GX
HF$GWY= 000010

. ABS.  000104    000   (RW,I,GBL,ABS,OVR)
        000330    001   (RW,I,LCL,REL,CON)
DATA    000212    002   (RW,D,LCL,REL,CON)
$STATE  000040    003   (RW,D,LCL,REL,CON)
$KTAB   000002    004   (RW,D,LCL,REL,CON)
$KSTR   000007    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13400 Words  ( 53 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:20.42
SY:VFPX29.V2,[132,134]VFPX29/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VFPX29
```

```
                                        .SBTTL  X3P$DF ACTION ROUTINES

                                ;
                                ; DEFAULT BLOCK SIZE
                                ;
000060                          BLKDEF: CALL    CHKBLK                  ; CHECK FOR LEGAL BLOCK SIZE VALUE
000064  103403                          BCS     101$                    ; BR IF ILLEGAL VALUE
000066  010067  000000G                 MOV     R0,BLKSZD               ; STORE DEFAULT BLOCK SIZE
000072                                  RETURN
                                ;
                                ; ERRORS
                                ;
000074                          101$:   EMSG$R  P7                      ; ERROR - ILLEGAL DEFAULT BLOCK SIZE

                                ;
                                ; MAXIMUM BLOCK SIZE
                                ;
000102                          BLKMAX: CALL    CHKBLK                  ; CHECK FOR LEGAL BLOCK SIZE VALUE
000106  103406                          BCS     101$                    ; BR IF ILLEGAL VALUE
000110  020067  000000G                 CMP     R0,BLKSZD               ; IS DEFAULT IN RANGE?
000114  103406                          BLO     102$                    ; BR IF NO
000116  010067  000000G                 MOV     R0,BLKSZM               ; STORE MAXIMUM BLOCK SIZE
000122                                  RETURN
                                ;
                                ; ERRORS
                                ;
000124                          101$:   EMSG$R  P8                      ; ERROR - ILLEGAL MAXIMUM BLOCK SIZE
000132                          102$:   EMSG$R  01                      ; ERROR - DEFAULT EXCEEDS MAXIMUM

                                ;
                                ; DEFAULT WINDOW SIZE
                                ;
000140                          WNDDEF: CALL    CHKWND                  ; CHECK FOR LEGAL WINDOW SIZE VALUE
000144  103403                          BCS     101$                    ; BR IF ILLEGAL VALUE
000146  110067  000000G                 MOVB    R0,WNDSZD               ; STORE DEFAULT WINDOW SIZE
000152                                  RETURN
                                ;
                                ; ERRORS
                                ;
000154                          101$:   EMSG$R  P9                      ; ERROR - ILLEGAL DEFAULT WINDOWSIZE

                                ;
                                ; MAXIMUM WINDOW SIZE
                                ;
000162                          WNDMAX: CALL    CHKWND                  ; CHECK FOR LEGAL WINDOW SIZE VALUE
000166  103406                          BCS     101$                    ; BR IF ILLEGAL VALUE
000170  020067  000000G                 CMP     R0,WNDSZD               ; IS DEFAULT IN RANGE?
000174  103406                          BLO     102$                    ; BR IF NO
000176  010067  000000G                 MOV     R0,WNDSZM               ; STORE MAXIMUM WINDOW SIZE
000202                                  RETURN
                                ;
                                ; ERRORS
                                ;
000204                          101$:   EMSG$R  00                      ; ERROR - ILLEGAL MAXIMUM WINDOW SIZE
000212                          102$:   EMSG$R  02                      ; ERROR - DEFAULT EXCEEDS MAXIMUM
```

```
VHDAT      CREATED BY  MACRO  ON 29-JUN-85 AT 02:42     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00
MACRO NAME       REFERENCES

FDBDF$           #5-68        5-68
FDBSZ$           #5-68        5-68
HLPDF$           #5-59        5-68
```

```
**FILE**ID**VIMI




VV       VV      IIIIII      MM       MM      IIIIII
VV       VV      IIIIII      MM       MM      IIIIII
VV       VV        II        MMMM   MMMM        II
VV       VV        II        MMMM   MMMM        II
VV       VV        II        MM  MM   MM        II
VV       VV        II        MM  MM   MM        II
VV       VV        II        MM       MM        II
VV       VV        II        MM       MM        II
VV       VV        II        MM       MM        II
VV       VV        II        MM       MM        II
  VV   VV          II        MM       MM        II             ....
  VV   VV          II        MM       MM        II             ....
     VV          IIIIII      MM       MM      IIIIII           ....
     VV          IIIIII      MM       MM      IIIIII           ....




LL               SSSSSSSS    TTTTTTTTTT
LL               SSSSSSSS    TTTTTTTTTT
LL             SS                TT
LL             SS                TT
LL             SS                TT
LL             SS                TT
LL               SSSSSS          TT
LL               SSSSSS          TT
LL                      SS       TT
LL                      SS       TT
LL                      SS       TT
LL                      SS       TT
LLLLLLLLLL     SSSSSSSS          TT
LLLLLLLLLL     SSSSSSSS          TT
```

```
**FILE**ID**VIMO




VV      VV     IIIIII     MM      MM      000000
VV      VV     IIIIII     MM      MM      000000
VV      VV       II       MMMM  MMMM    00      00
VV      VV       II       MMMM  MMMM    00      00
VV      VV       II       MM  MM  MM    00      00
VV      VV       II       MM  MM  MM    00      00
VV      VV       II       MM      MM    00      00
VV      VV       II       MM      MM    00      00
VV      VV       II       MM      MM    00      00
VV      VV       II       MM      MM    00      00
  VV  VV         II       MM      MM    00      00       ....
  VV  VV         II       MM      MM    00      00       ....
    VV         IIIIII     MM      MM      000000         ....
    VV         IIIIII     MM      MM      000000         ....




LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
    67                                          ;+
    68                                          ;       **-$VINI
    69                                          ;
    70                                          ; INPUTS:
    71                                          ;       NONE
    72                                          ;
    73                                          ; OUTPUTS:
    74                                          ;       SYSTEM IMAGE FILE OPENED AND VERIFIED
    75                                          ;       SYMBOLS DEFINED FROM STB FILE
    76                                          ;       ALL REGISTERS DESTROYED
    77                                          ;-
    78
    79                                                  .MCALL  HWDDF$,LBLDF$,PCBDF$,SWTCHO,ERMSG$,ERBLK$,ERRPT$,SWTCHI,GETRV
    80                                                  .MCALL  CALLR
    81                                                  .ENABL  LC                      ; ENABLE LOWER CASE
    82
    83 000006                                           HWDDF$                          ; HARDWARE DEFINITIONS
    84 000006                                           LBLDF$                          ; LABEL BLOCK OFFSETS
    85 000006                                           PCBDF$                          ; PARTITION CONTROL BLOCK OFFSETS
    86
    87 000006                                   $VINI::
    88 000006                                           CALL    VFILE                   ; OPEN SYSTEM IMAGE FILE
    89 000012  103431                                   BCS     100$                    ; EXIT GRACEFULLY
    90 000014                                           CALL    VSYMB                   ; OPEN SYMBOL TABLE FILE AND DEFINE SYMBOLS
    91 000020  103426                                   BCS     100$                    ; CAREFULLY CAREFULLY
    92 000022                                           CALL    VSYM1                   ; UPDATE SYMBOL TABLE
    93 000026  103423                                   BCS     100$                    ; ERROR
    94
    95                                                  .IF DF  R$$MPL                                          ;[MP01]
    96                                                  BIT     #DF.DAS,$DFLAG  ; D-SPACE PROBLEM ?                     ;[MP01]
    97                                                  BEQ     10$             ; IF NO - BRANCH                        ;[MP01]
    98                                                  MOV     #$ICBD,$ICBLN   ; IF YES - UPDATE ICB SIZE              ;[MP01]
    99
   100                                                  .ENDC                                                   ;[MP01]
   101
   102 000030                                   10$:    CALL    $XXINI          ; INITIALIZE OND OF TASK BUFFER
   103 000034  103421                                   BCS     110$            ; GET TASK DIRECTIVE NOT SUPPORTED
   104 000036  016767  000000G 000000G                  MOV     $XXBUF,$XBUF2   ; INITIALIZE NETLOADER BUFFER POINTERS
   105 000044  016767  000000G 000000G                  MOV     $XXLEN,$XLEN2   ; ...
   106 000052  052767  000000G 000000G                  BIS     #VF.INI,$VFLAG  ; INDICATE SUCCESSFUL INITIALIZATION
   107
   108 000060  032767  020000  000000G                  BIT     #FE.CEX,$FMASK  ; IS COMEXEC LOADED ?
   109 000066  001403                                   BEQ     100$            ; IF EQ, NO
   110 000070  005002                                   CLR     R2              ; NO RELOCATION BIAS
   111 000072                                           CALL    $VIMI           ; LOAD CETAB IMAGE, ETC.
   112 000076                                   100$:   RETURN
   113
   114                                          ; ERROR TEXT
   115
   116 000100                                   110$:   ERRPT$  ERROR1,$EROUT,FATAL     ; GTSK$S DIRECTIVE NOT SUPPORTED
   117 000114                                           ERMSG$  <GTSK$S directive not supported by operating system.>
   118 000177                                           ERBLK$  ERROR1
```

```
                .WORD   0
        URMTB:  .WORD   STMFC,0         ; UNIBUS RUN CONECTIVITY TABLE
                .RAD50  "$URMTB"
                .WORD   0
        STMFC:  .WORD   LTMFC,0         ; 100 MSEC TIMEOUT FUNCTION CODE
                .RAD50  "$STMFC"
                .WORD   0
        LTMFC:  .WORD   CCBAF,0         ; LONG TIMEOUT FUNCTION CODE
                .RAD50  "$LTMFC"
                .WORD   0
        CCBAF:  .WORD   RDBAF,0         ; ALLOCATION FAILURES
                .RAD50  "$CCBAF"
                .WORD   0
        RDBAF:  .WORD   LDBAF,0         ; ...
                .RAD50  "$RDBAF"
                .WORD   0
        LDBAF:  .WORD   SDBAF,0         ; ...
                .RAD50  "$LDBAF"
                .WORD   0
        SDBAF:  .WORD   CCBAL,0         ; ...
                .RAD50  "$SDBAF"
                .WORD   0
        CCBAL:  .WORD   CMFRK,0         ; NUMBER OF DYNAMIC CCB ALLOCATIONS
                .RAD50  "$CCBAL"
                .WORD   0
        CMFRK:  .WORD   CEX,0           ; FORK BLOCK LISTHEAD
                .RAD50  "$CMFRK"
                .WORD   0
        CEX:    .WORD   OBJHD,0         ; CEX ADDRESS IN RSX EXEC SPACE
                .RAD50  "$CEX"
                .WORD   0
        OBJHD:  .WORD   FILHD,0         ; OBJECT LISTHEAD                       ;[MP01]
                .RAD50  "$OBJHD"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        FILHD:  .WORD   CCBLH,0         ; FILTER LISTHEAD                       ;[MP01]
                .RAD50  "$FILHD"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        CCBLH:  .WORD   SDBLH,0         ; CCB LISTHEAD                          ;[MP01]
                .RAD50  "$CCBLH"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        SDBLH:  .WORD   RDBLH,0         ; SDB LISTHEAD                          ;[MP01]
                .RAD50  "$SDBLH"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        RDBLH:  .WORD   SKS5,0          ; LDB LISTHEAD                          ;[MP01]
                .RAD50  "$RDBLH"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        SKS5:   .WORD   NTLPT,0         ; S.KS5 SCB OFFSET                      ;[MP01]
                .RAD50  "S.KS5"                                                 ;[MP01]
                .WORD   0                                                       ;[MP01]
        NTLPT:  .WORD   PSIPT,0         ; NTL HOME BLOCK                        ;[MP01]
                .RAD50  "$NTLPT"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        PSIPT:  .WORD   POLHD,0         ; PSI HOME BLOCK                        ;[MP01]
                .RAD50  "$PSIPT"                                                ;[MP01]
                .WORD   0                                                       ;[MP01]
        POLHD:  .WORD   SECFR,0         ; SECONDARY POOL LISTHEAD               ;[MP01]
                .RAD50  "$POLHD"                                                ;[MP01]
```

```
VINI        CREATED BY  MACRO  ON 29-JUN-85 AT 02:43     PAGE 4
SYMBOL  CROSS REFERENCE                                  CREF    04.00

SYMBOL    VALUE             REFERENCES

$VCTOF   = ******   GX     *11-591
$VCTVB   = ******   GX     *11-590
$VFLAG   = ******   GX     *7-106     *8-160     *9-442
$VFLG1   = ******   GX     *8-252
$VGCL    = ******   GX      8-145
$VIMI    = ******   GX      7-111
$VINI      000006 RG       #7-87
$XBUF2   = ******   GX     *7-104
$XFRAD   = ******   GX     *8-217
$XLEN2   = ******   GX     *7-105
$XXBUF   = ******   GX      7-104
$XXINI   = ******   GX      7-102
$XXLEN   = ******   GX      7-105
$$       = 177777           7-117      7-117     #7-117     #7-118      8-260      8-260     #8-260     #8-261      8-262
                            8-262     #8-262     #8-263      8-264      8-264     #8-264      8-265      8-265     #8-265
                           #8-266      8-267      8-267     #8-267      8-268      8-268     #8-268     #8-269      8-270
                            8-270     #8-270      8-271      8-271     #8-271     #8-272      9-458      9-458     #9-458
                            9-459      9-459     #9-459     #9-460      9-461      9-461     #9-461      9-462      9-462
                           #9-462     #9-463      10-528     10-528    #10-528    #10-529     10-530     10-530    #10-530
                           #10-531     11-669     11-669    #11-669    #11-670     13-999     13-999    #13-999    #13-1000
                            13-1001    13-1001   #13-1001   #13-1002    13-1003    13-1003   #13-1003   #13-1004    13-1005
                            13-1005   #13-1005    13-1006    13-1006   #13-1006   #13-1007
$$$      = 004604 R        #7-117      7-118     #8-260      8-261     #8-262      8-263     #8-264      8-266     #8-267
                            8-269     #8-270      8-272     #9-458      9-460     #9-461      9-463     #10-528     10-529
                           #10-530     10-531    #11-669     11-670    #13-999     13-1000   #13-1001    13-1002   #13-1003
                            13-1004   #13-1005    13-1007
$$$$     = 004754 R        #7-118     #8-261     #8-263     #8-266     #8-269     #8-272     #9-460     #9-463     #10-529
                           #10-531    #11-670     11-671    #13-1000   #13-1002   #13-1004   #13-1007
.COO       003322 R         12-805    #12-808
.CRAVL   = ******   GX      11-626
.CRCOV   = ******   GX      11-657    *11-659
.CSI1    = ******   G       8-162      8-162
.CSI2    = ******   G       8-164      8-164
.DEVHD   = ******   GX      11-621
.DLCOV   = ******   GX      11-654    *11-656
.EXSIZ   = ******   GX      11-614
.FMASK   = ******   GX      9-437      11-612
.NTUIC   = ******   GX     *9-394      9-395      9-397
.OPEN    = ******   G       8-168      8-168
.PARHD   = ******   GX      11-623
.PLGTH   = ******   GX      13-948
.PPASC   = ******   GX      14-1025
.RDFUI   = ******   GX      14-1021
.SYSIZ   = ******   GX      11-586
```

```

                             ;
                             ; Macro calls and local data
                             ;
                                     .MCALL  VNPDF$,DIR$,QIOW$,UCBDF$,GETRV,SAVRG,RESRG,ASL$,DHBDF$,SWTCHI
                                     .ENABL  LC

000000                               VNPDF$                  ; Netldr request block offsets
000000                               UCBDF$                  ; Define ucb offsets
000000                               DHBDF$                  ; DEFINE DECNET HOME BLOCK OFFSETS


          000077                     MXAREA = 63.            ; Maximum area
          001777                     MXADDR = 1023.          ; Maximum node address
          000006                     MXALIN = 6.             ; Maximum alias name length
          000077                     MXCOP  = 63.            ; Maximum number of object task copies
          000031                     MXCST  = 25.            ; Maximum line cost value
          000031                     MXCST  = 25.            ; Maximum line cost value
          000006                     MXDSTN = 6.             ; Maximum destination name length [old format]   ;[TD001]
          000020                     MXDTEL = 16.            ; Maximum DTE address length                     ;[TD001]
          000006                     MXGRPN = 6.             ; Maximum group name length [old format]         ;[TD001]
          000006                     MXPSCN = 6.             ; Maximum PSI-circuit name length [old format]   ;[TD001]
          000050                     MXSNKN = 40.            ; Maximum sink name length                       ;[TD001]
          000006                     MXNODN = 6.             ; Maximum node name length.                      ;[TD001]
          000020                     MAXID  = 16.            ; Max length of net. mgmt. 'id-string'           ;[TD001]
          000040                     SPACE  = 40             ; ASCII space character                          ;[TD001]
                             ;
                             ; Define error message text
                             ;
                                     .MACRO  ETEXT$  MSG,TEXT
                             MSG:
                                     .ASCII  "VNP -- TEXT"
                             L'MSG'=.-MSG
                                     .ENDM   ETEXT$


                             ;
                             ; Print error message
                             ;
                                     .MACRO  ERROR$  MSG,WARN
                                     MOV     #MSG,EBUF       ; Set dpb buffer address
                                     MOV     #L'MSG',ELEN    ;  and buffer length
                                     .IF NB  WARN
                                     CALL    ERRHDL          ; Call the error handler
                                     .IFF
                                     JMP     ERRHDL          ; Jump to the error handler
                                     .ENDC
                                     .ENDM   ERROR$

000000                               .PSECT  DATA,D
                             ;
                             ; Error message dpb
                             ;
000000                       EMSG:   QIOW$   IO.WLB,1,1,,,,<,,40>
          000014'            EBUF=EMSG+Q.IOPL
          000016'            ELEN=EBUF+2
                             ;
                             ; General purpose dpb (no vertical foratting
002540  001401                             BEQ     101$            ; If zero, illegal value
002542                                     RETURN
002544                             101$:   ERROR$  ILLINA          ; Illegal inactivity timer value
                                   ;
                                   ; Retransmit factor
                                   ;
002564  005767  000000G            $RET::  TST     .PNUMH          ; Must be a word value
002570  001010                             BNE     101$            ; If ne, illegal value
002572  105767  000001G                    TSTB    .PNUMB+1        ; Must be a byte value
002576  001005                             BNE     101$            ; If ne, illegal value
002600  016767  000000G 000064G            MOV     .PNUMB,RQB+LO.RET ; Store retransmit factor
002606  001401                             BEQ     101$            ; If zero, illegal value
002610                                     RETURN
002612                             101$:   ERROR$  ILLRET          ; Illegal retransmit factor value
                                   ;
                                   ; Controller table count
                                   ;
002632  005767  000000G            $MXCO:: TST     .PNUMH          ; Double precision?
002636  001010                             BNE     101$            ; If ne, yes - too big
002640  016700  000000G                    MOV     .PNUMB,R0       ; Get line table count
002644  022700  000100                     CMP     #64.,R0         ; Is the count too big?
002650  103403                             BLO     101$            ; If lo, yes
002652  010067  000006G                    MOV     R0,RQB+LO.CON   ; Set controller increment value
002656                                     RETURN
002660                             101$:   ERROR$  MXCN64


                                   ;
                                   ; Line table count
                                   ;
002700  005767  000000G            $MXLI:: TST     .PNUMH          ; Double precision?
002704  001010                             BNE     101$            ; If ne, yes - too big
002706  016700  000000G                    MOV     .PNUMB,R0       ; Get line table count
002712  022700  000100                     CMP     #64.,R0         ; Is the count too big?
002716  103403                             BLO     101$            ; If lo, yes
002720  010067  000004G                    MOV     R0,RQB+LO.LIN   ; Set line increment value
002724                                     RETURN
002726                             101$:   ERROR$  MXLN64



                                   ;
                                   ; Store node area/address
                                   ;
002746  005067  000100'            $ARAD1::CLR     AREAN           ; assume default area
002752  016767  000000G 000102'            MOV     .PNUMB,ADDRS    ; store area or address, not determined yet
002760                                     RETURN

002762  016700  000102'            $ARAD2::MOV     ADDRS,R0        ; store area number
002766  005700                             TST     R0              ; check for area lower limit
002770  001424                             BEQ     101$            ; if illegal - branch
002772  022700  000077                     CMP     #MXAREA,R0      ; check for high limit
002776  103421                             BLO     101$            ; if illegal - branch
003000  000241                             CLC                     ; do not rol c-bit

        000012                             .REPT   10.             ; position area bits
                                           ROL     R0              ;  ...
                                           .ENDM

```

```
                                        ;
                                        ; Help action routine
                                        ;
006276  010467  000002G                 $HLPAC::MOV     R4,RQB+LR.HLP   ; Save address of help string
006302                                          RETURN

                                        ;
                                        ; Error hanlder
                                        ;
006304                                  ERRHDL: DIR$    #EMSG           ; Print error message
006312                                          RETURN

```

```
                                                   .SBTTL  INIXCB - INITIALIZE X25 CIRCUIT BLOCK
                                          ;+
                                          ;
                                          ; INIXCB - INITIALIZE X25 CIRCUIT BLOCK
                                          ;
                                          ; INPUTS:
                                          ;        R0 - UNMAPPED ADDRESS OF CIRCUIT BLOCK
                                          ;        $PFLAG - PSI characteristics word                                    ;[TD001]
                                          ;
                                          ; OUTPUTS:
                                          ;        R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK
                                          ;        X25 CIRCUIT BLOCK INITIALIZED ACCORDING TO VALUES
                                          ;                SPECIFIED IN PVC$DF
                                          ;        R0,R1,R2 DESTROYED
                                          ;
                                          ;-
001676                                    INIXCB: CEACC$  R0,R3            ; GET MAPPED ADDRESS OF CIRCUIT BLOCK
001706  012701  000106                            MOV     #X$LEN,R1        ; GET SIZE OF CIRCUIT BLOCK
001712  032767  000000G 000000G                   BIT     #PF.PSV,$PFLAG   ; Check for PSV module                               ;[TD001]
001720  001402                                    BEQ     5$               ; Branch if not present                              ;[TD001]
001722  012701  000106                            MOV     #X$GLEN,R1       ; PSV needs longer XCB                               ;[TD001]
001726                                    5$:     GETAD   R3,R1            ; READ IN CIRCUIT BLOCK                              ;[TD001]
001742  012702  000000G                           MOV     #$BFR,R2         ; GET CIRCUIT BLOCK ADDRESS                          ;**-1
001746  105022                            10$:    CLRB    (R2)+            ; INITIALIZE CIRCUIT BLOCK TO ZEROES
001750  005301                                    DEC     R1               ; MORE TO INITIALIZE?
001752  003375                                    BGT     10$              ; BR IF YES
001754  010300                                    MOV     R3,R0            ; COPY VIRTUAL ADDRESS OF XCB
001756                                            CALL    BIASX            ; BIAS CORRECTLY FOR APR6
001762  016767  000012' 000020G                   MOV     PVCWSZ,X$WSZ+$BFR ; STORE WINDOW SIZE
001770  016767  000010' 000016G                   MOV     PVCPSZ,X$PKSZ+$BFR; STORE PACKET SIZE
001776  016767  000006' 000060G                   MOV     PVCCTM,X$CTIM+$BFR; STORE COUNTER TIMER
002004  016767  000000G 000024G                   MOV     DTEDES,X$DTE+$BFR ; STORE DTE DESCRIPTOR ADDRESS
002012  116767  000024' 000021G                   MOVB    PVCPRT,X$PRT+$BFR ; STORE PORT NUMBER
002020  016767  000004' 000026G                   MOV     PVCLCN,X$LCN+$BFR ; STORE LOGICAL CHANNEL NUMBER
002026  010067  000070G                           MOV     R0,X$TXQ+2+$BFR    ; INITIALISE TRANSMISSION QUEUE
002032  062767  000066  000070G                   ADD     #X$TXQ,X$TXQ+2+$BFR ; ...
002040  010067  000074G                           MOV     R0,X$RXQ+2+$BFR    ; INITIALISE RECEIVED DATA QUEUE
002044  062767  000072  000074G                   ADD     #X$RXQ,X$RXQ+2+$BFR ; ...
002052  010067  000064G                           MOV     R0,X$WAQ+2+$BFR    ; INITIALISE ACKNOWLEDGEMENT QUEUE
002056  062767  000062  000064G                   ADD     #X$WAQ,X$WAQ+2+$BFR ; ...
002064  010067  000100G                           MOV     R0,X$ABQ+2+$BFR    ; INITIALISE ABORTED DATA QUEUE
002070  062767  000076  000100G                   ADD     #X$ABQ,X$ABQ+2+$BFR ; ...
002076  116767  000054' 000005G                   MOVB    SP$P4,X$ST+$BFR    ; SET PORT IN DATA TRANSFER STATE
002104  112767  000001  000015G                   MOVB    #XT$PVC,X$TYP+$BFR ; SET CIRCUIT TYPE AS PVC
002112                                            PUTAD                      ; WRITE UPDATED CIRCUIT BLOCK
002116                                            RETURN
```

```
     55                                         .SBTTL  MACRO DEFINITIONS
     56
     57                                 ;
     58                                 ; LIBRARY MACROS
     59                                 ;
     60                                         .MCALL  RDTDF$,EMSG$R,PHBDF$,NTLER$,STATE$,ISTAT$,TRAN$
     61                                         .MCALL  GETAD,PUTAD,GETRV,CEACC$
     62
     63 000000                                  PHBDF$                          ; PSI HOME BLOCK OFFSETS
     64 000000                                  RDTDF$                          ; REMOTE DTE NAME BLOCK OFFSETS
     65
     66                                 ;
     67                                 ; LOCAL MACRO DEFINITIONS
     68                                 ;
     69                                         .MACRO  SAVMAP
     70                                         MOV     $RBLCK,-(SP)
     71                                         .ENDM
     72
     73                                         .MACRO  RESMAP
     74                                         MOV     (SP)+,$RBLCK
     75                                         .ENDM
     76
```

```
**FILE**ID**VFPRNW





VV      VV   FFFFFFFFFF   PPPPPPPP     RRRRRRRR     NN      NN   WW      WW
VV      VV   FFFFFFFFFF   PPPPPPPP     RRRRRRRR     NN      NN   WW      WW
VV      VV   FF           PP      PP   RR      RR   NN      NN   WW      WW
VV      VV   FF           PP      PP   RR      RR   NN      NN   WW      WW
VV      VV   FF           PP      PP   RR      RR   NNNN    NN   WW      WW
VV      VV   FF           PP      PP   RR      RR   NNNN    NN   WW      WW
VV      VV   FFFFFFFF     PPPPPPPP     RRRRRRRR     NN  NN  NN   WW      WW
VV      VV   FFFFFFFF     PPPPPPPP     RRRRRRRR     NN  NN  NN   WW      WW
VV      VV   FF           PP           RR  RR       NN    NNNN   WW  WW  WW
VV      VV   FF           PP           RR  RR       NN    NNNN   WW  WW  WW
  VV  VV     FF           PP           RR    RR     NN      NN   WWWW  WWWW       ....
  VV  VV     FF           PP           RR    RR     NN      NN   WWWW  WWWW       ....
    VV       FF           PP           RR      RR   NN      NN   WW      WW       ....
    VV       FF           PP           RR      RR   NN      NN   WW      WW       ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                      ;+
                                      ; *** RNWNOD - Check node name
                                      ;
                                      ; INPUTS
                                      ;       .PSTPT - Pointer to parsed string
                                      ;       .PSTCN - Number of characters in string
                                      ;
                                      ; OUTPUTS
                                      ;       Node name checked and stored
                                      ;       Error condition if invalid node name
                                      ;-
000144                                RNWNOD:
000144  016701  000000G                       MOV     .PSTCN,R1
000150  020127  000006                        CMP     R1,#MXNOID              ; Check if node name too long
000154  101011                                BHI     101$                    ; Branch if error
000156  110167  000000G                       MOVB    R1,RNNL                 ; Move node length to data area
000162  016700  000000G                       MOV     .PSTPT,R0               ; Move node name to data area
000166  012702  000000G                       MOV     #RNNODE,R2
000172                                        CALL    MOVE
000176                                        RETURN

000200                                101$:   EMSG$R  RB                      ; Illegal node name
```

```
VFPRN

SYMBO

SYMBO

R$NOD
R$NWK
R$NWL
R$PAL
R$PAS
R$USL
R$USR
SPFIL
STORC
STRBU
STRLE
SYNER
SYSFD
S$$BA
$ALPH
$ALPO
$ANY
$BFR

$BLAN
$CEAC
$DBG$
$DIGI
$DNUM
$EOS
$ERRR
$ERRR
$ERRR
$ERRR
$ERR1
$EXIT
$FAIL
$GETA
$GETR
$GPRM
$LAMD
$NUMB
$PSIP
$PUTA
$QRNA
$QRNW
$RADD
$RAD5
$RBLC
$RONL
$SAVA
$STRN
$SUBX
$$$FL
$$$KE
.PCHA
.PSTC
```

```
                                                              L  5
VFPRNW      CREATED BY  MACRO  ON 29-JUN-85 AT 02:40      PAGE 2

SYMBOL  CROSS REFERENCE                                   CREF    04.00

SYMBOL   VALUE                REFERENCES

R$NOD      000023        21-466
R$NWK      000003        20-426     21-464
R$NWL      000104        20-422     21-451     21-457     21-488
R$PAL      000052       *21-480
R$PAS      000053        21-481
R$USL      000031       *21-475
R$USR      000032        21-476
SPFILL   = ****** GX     21-468
STORCH     001020 R     #23-519
STRBUF     000165 R     #8-101      16-325     17-348     18-372    *23-522
STRLEN     000164 R     #8-100      16-321     17-344     18-368    *22-505     23-519     23-521    *23-523
SYNERR   = ****** GX    *9-130     *9-133
SYSFDB   = ****** GX    *20-417    *20-422    *21-488
S$$BAS   = ******        7-90       7-90       7-91       7-91       7-92       7-92       7-93       7-93
$ALPHA   = 000022       #10-144
$ALPOL   = ****** GX     21-452
$ANY     = 000020       #10-144
$BFR     = ****** GX     20-418     20-423     20-426     21-456    *21-463     21-464     21-466    *21-475     21-476
                        *21-480     21-481    *21-485     21-486

$BLANK   = 000006       #10-144
$CEACC   = ****** GX     20-421     21-455
$DBG$    = ******        9-125
$DIGIT   = 000024       #10-144
$DNUMB   = 000014       #10-144
$EOS     = 000012       #10-144
$ERRRA     000000 R     #7-90       14-233     19-397
$ERRRB     000026 R     #7-91       15-306
$ERRRC     000050 R     #7-92       16-330     17-353     18-377
$ERRRX     000104 R     #7-93       19-398
$ERR1T   = ****** GX     9-139
$EXIT    = 000000       #10-144
$FAIL    = 177777       #10-144
$GETAD   = ****** GX     20-422
$GETRV   = ****** GX     20-417
$GPRM    = ******        10-144
$LAMDA   = 000000       #10-144
$NUMBR   = 000002       #10-144
$PSIPT   = ****** GX     20-415     21-489
$PUTAD   = ****** GX     21-488
$QRNA      000006 RG    #9-123
$QRNW      000000 RG    #9-120
$RADDR   = ****** GX    *20-417    *20-422    *21-488
$RAD50   = 000016       #10-144
$RBLCK   = ****** GX     20-414    *20-432     21-450    *21-492
$RONLY   = ******        10-144     10-144     10-144
$SAVAL   = ****** GX     21-449
$STRNG   = 000004       #10-144
$SUBXP   = 000010       #10-144
$$$FLG   = 177777       #10-144
$$$KEY   = 177777       #10-144
.PCHAR   = ****** GX     23-522
.PSTCN   = ****** GX     14-270     15-297
```

```
VFPX29      CREATED BY  MACRO  ON 29-JUN-85 AT 02:41      PAGE 1
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL    VALUE              REFERENCES

CERR      = ******   GX       6-74       6-75       6-76       6-77
CFGBF     = ******   GX       7-107
CFGSZ     = ******   GX       7-106
CFLIN     = ******   GX       6-74       6-75       6-76       6-77
CTIMR       000210 R         #6-86      *9-170      9-195
FMT10     = ******   GX       6-74       6-75       6-76
FMT8      = ******   GX       6-77
FM.10     = 000000           #6-74      #6-75      #6-76
FM.8      = 000000           #6-77
F.RSIZ    = ******   GX      *9-181     *9-197     *9-200
H$X29C      000040            9-180      9-199
MAXVC       000206 R         #6-85      *9-160      9-194
Q$CTIM      000000           *9-195
Q$LEN       000016            9-185      9-189      9-197
Q$MXVC      000002           *9-194
Q$TCLZ      000010           *9-196
RTSPC     = ******   GX       6-74       6-75       6-76       6-77
SYNERR    = ******   GX      *7-108     *7-111
SYSFDB    = ******   GX      *9-181     *9-197     *9-200
S$$BAS    = ******            6-74       6-74       6-75       6-75       6-76       6-76       6-77       6-77
X29CTM      000104 R         #9-168
X29DF       000000 R          7-103
X29END      000130 R         #9-178
X29KW       000000 RG         7-105     #8-124
X29MXC      000052 R         #9-156
X29ST       000000 RG        #8-124
$ALL16    = ******   GX       9-186
$ALPHA    = 000022           #8-124
$ANY      = 000020           #8-124
$BFR      = ******   GX       9-182      9-188     *9-194     *9-195     *9-196     *9-201
$BLANK    = 000006           #8-124
$DIGIT    = 000024           #8-124
$DNUMB    = 000014           #8-124
$EOS      = 000012           #8-124
$ERR1T    = ******   GX       7-118
$ERR9A      000000 R         #6-74       9-205
$ERR9B      000036 R         #6-75       9-163
$ERR9C      000076 R         #6-76       9-173
$ERR9D      000132 R         #6-77       9-206
$EXIT     = 000000           #8-124
$FAIL     = 177777           #8-124
$GETRV    = ******   GX       9-181      9-200
$GPRM     = ******            8-124
$LAMDA    = 000000           #8-124
$NUMBR    = 000002           #8-124
$PSIPT    = ******   GX       9-178      9-198
$PUTRC    = ******   GX       9-197      9-202
$QX29       000000 RG        #7-103
$RADDR    = ******   GX      *9-181     *9-197     *9-200
$RAD50    = 000016           #8-124
$RONLY    = ******            8-124      8-124      8-124
$STRNG    = 000004           #8-124
```

```
                                                  .SBTTL  CHN$DF ACTION ROUTINES

                                        ;
                                        ; LOW END OF CHANNEL RANGE
                                        ;
                                                  .ENABL  LSB
000220  005767  000000G                 CHLO:     TST     .PNUMH              ; IS THIS A LEGAL VALUE?
000224  001157                                    BNE     101$                ; BR IF NO
000226  016700  000000G                           MOV     .PNUMB,R0           ; GET SPECIFIED CHANNEL RANGE
000232  003554                                    BLE     101$                ; BR IF ILLEGAL VALUE
000234  022700  000000G                           CMP     #CHNLMX,R0          ; LEGAL VALUE?
000240  103551                                    BLO     101$                ; BR IF NO
000242  010067  000000'                           MOV     R0,CHNLLO           ; SAVE LOW END VALUE
000246  010067  000002'                           MOV     R0,CHNLHI           ; THIS IS DEFAULT IF HIGH END VALUE OMITTED
000252                                            RETURN

                                        ;
                                        ; HIGH END OF CHANNEL RANGE
                                        ;
000254  005767  000000G                 CHHI:     TST     .PNUMH              ; IS THIS A LEGAL VALUE?
000260  001141                                    BNE     101$                ; BR IF NO
000262  016700  000000G                           MOV     .PNUMB,R0           ; GET SPECIFIED CHANNEL RANGE
000266  003536                                    BLE     101$                ; BR IF ILLEGAL VALUE
000270  022700  000000G                           CMP     #CHNLMX,R0          ; LEGAL VALUE?
000274  103533                                    BLO     101$                ; BR IF NO
000276  010067  000002'                           MOV     R0,CHNLHI           ; SAVE HIGH END VALUE
000302  012701  000006                  CHEND:    MOV     #K$LEN,R1           ; GET NUMBER OF BYTES TO ALLOCATE

000306                                            CALL    $SAVAL              ; SAVE ALL REGISTERS
000312                                            CALL    $ALL16              ; ALLOCATE CHANNEL LIST FROM EXEC POOL
000316  103525                                    BCS     102$                ; BR IF UNSUCCESSFUL ALLOCATION
000320  005067  000000G                           CLR     $BFR                ; INITIALIZE LINK WORD
000324  016767  000000' 000002G                   MOV     CHNLLO,K$LOC+$BFR;  STORE LOW END OF CHANNEL RANGE
000332  016767  000002' 000004G                   MOV     CHNLHI,K$HIC+$BFR;  STORE HIGH END OF CHANNEL RANGE
000340                                            PUTRC   R0,#K$LEN           ; WRITE UPDATED CHANNEL BLOCK
000356                                            SAVMAP                      ; SAVE CURRENT MAPPING
000362                                            CEACC$  DTEDES,R1           ; CONVERT TO MAPPED ADDRESS
000374  010102                                    MOV     R1,R2               ; SAVE IT
000376                                            GETAD   R1,#L$LEN           ; READ IN DTE BLOCK
000414  016767  000034G 000000G                   MOV     L$CHLS+$BFR,$BFR;   GET FIRST CHANNELS LIST
000422  001014                                    BNE     20$                 ; BR IF NOT EMPTY
000424  010067  000000G                           MOV     R0,$BFR             ; GET ADDRESS OF CHANNEL BLOCK
000430  062701  000034                            ADD     #L$CHLS,R1          ; POINT TO CHANNELS LISTHEAD
000434                                            PUTAD   R1,#2               ; REWRITE CHANNELS LISTHEAD
000452  000417                                    BR      25$                 ; CONTINUE
000454                                  20$:      GETRV   $BFR,#K$LEN         ; READ IN CHANNELS BLOCK
000474  005767  000000G                           TST     $BFR                ; END OF LIST?
000500  001365                                    BNE     20$                 ; BR IF NOT END OF LIST
000502  010067  000000G                           MOV     R0,$BFR             ; ADD BLOCK TO CHANNELS LIST
000506                                            PUTRC                       ; REWRITE CHANNELS BLOCK

000512  016701  000002'                 25$:      MOV     CHNLHI,R1           ; GET HIGH END OF CHANNELS
000516  166701  000000'                           SUB     CHNLLO,R1           ; COMPUTE HIGH MINUS LOW
000522  002001                                    BGE     30$                 ; IF GE, RESULT CORRECT
000524  005401                                    NEG     R1                  ; ELSE, GET ACTUAL DIFFERENCE
000526  005201                          30$:      INC     R1                  ; GET NUMBER OF CHANNELS
000530                                            GETAD   R2,#L$LEN           ; READ IN DTE DESCRIPTOR
```

```
**FILE**ID**VHELP
```

```
VV      VV   HH      HH   EEEEEEEEEE   LL           PPPPPPPP
VV      VV   HH      HH   EEEEEEEEEE   LL           PPPPPPPP
VV      VV   HH      HH   EE           LL           PP      PP
VV      VV   HH      HH   EE           LL           PP      PP
VV      VV   HH      HH   EE           LL           PP      PP
VV      VV   HH      HH   EE           LL           PP      PP
VV      VV   HHHHHHHHHH   EEEEEEEE     LL           PPPPPPPP
VV      VV   HHHHHHHHHH   EEEEEEEE     LL           PPPPPPPP
VV      VV   HH      HH   EE           LL           PP
VV      VV   HH      HH   EE           LL           PP
  VV  VV     HH      HH   EE           LL           PP                ....
  VV  VV     HH      HH   EE           LL           PP                ....
    VV       HH      HH   EEEEEEEEEE   LLLLLLLLLL   PP                ....
    VV       HH      HH   EEEEEEEEEE   LLLLLLLLLL   PP                ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                .TITLE  VIMI - VNP READ IN CEX IMAGE
                .IDENT  /V05.00/
                .ENABLE LC

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       VNP - READ CEX IMAGES FROM DISK
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                        .TITLE  VIMO - VNP WRITE IMAGE OUT ROUTINES
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ; MODULE DESCRIPTION:
                ;
                ;       VNP - WRITE CEX IMAGES BACK TO DISK
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  27-FEB-78
                ;       VERSION 2.0 RELEASE
                ;
                ; 2.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;

```

```
                                        ;+
                                                **-VFILE

                                                OPEN SYSYEM IMAGE FILE

                                        ; INPUTS:
                                                NONE

                                        ; OUTPUTS:
                                                OPEN SYSTEM IMAGE FILE
                                                /OR/
                                                ERROR MESSAGE AND EXIT WITH C-BIT SET
                                        ;-

                                                .MCALL  OPEN$M,FDAT$R,FDRC$R,CSI$1,CSI$2,CLOSE$,GETRC,FDOFF$
                                                .MCALL  GCLDF$,PUTRC,CSI$SW,GETRV

 000202                                         FDOFF$  DEF$L           ; DEFINE FDB OFFSETS
 000202                                         GCLDF$                  ; DEFINE GCL OFFSETS

 000202 016746  000000G                 VFILE:  MOV     $CLPMT,-(SP)    ; SAVE CURRENT PROMPT STRING ADDRESS
 000206 016746  000000G                         MOV     $CLBUF,-(SP)    ; SAVE CURRENT BUFFER IN USE BY GCL
 000212 012767  001314' 000000G                 MOV     #SYSPMT,$CLPMT  ; PROMPT FOR FILENAME
 000220 012767  000000G 000000G                 MOV     #FILE,$CLBUF    ; READ INTO THIS BUFFER
 000226 042767  000001  000000G                 BIC     #CL.CMT,$CLFLG  ;*; ALLOW FOR ";" IN FILENAME SPEC.
 000234                                         CALL    $VGCL           ; GET COMMAND LINE
 000240 012667  000000G                         MOV     (SP)+,$CLBUF    ; RESTORE BUFFER ADDRESS
 000244 012667  000000G                         MOV     (SP)+,$CLPMT    ; RESTORE PROMPT STRING ADDRESS
 000250 103001                                  BCC     1$              ; CONTINUE GRACEFULLY                   ;[MP01]
 000252                                         RETURN                  ; EXIT GRACEFULLY                       ;[MP01]

 000254 052767  000001  000000G 1$:             BIS     #CL.CMT,$CLFLG  ;*; ALLOW COMMENTS AGAIN
 000262 105767  000000G                         TSTB    $CLEVL          ; ARE WE AT LEVEL 0 ?
 000266 001003                                  BNE     5$              ; NO
 000270 042767  100000  000000G                 BIC     #CL.XIT,$CLFLG  ; DON'T EXIT AFTER MCR COMMAND

 000276 016700  000000G         5$:             MOV     $CLIOS,R0       ; ADDRESS OF I/O STATUS BLOCK
 000302 012701  000000G                         MOV     #FILE,R1        ; ADDRESS OF BUFFER
 000306 016002  000002                          MOV     2(R0),R2        ; LENGTH OF FILENAME

 000312 042767  000000G 000000G                 BIC     #VF.SS,$VFLAG   ; ASSUME NOT AN RSX11S SYSTEM
 000320 012700  000000G                         MOV     #$CLCSI,R0      ; ADDRESS OF CSI CONTROL BLOCK
 000324                                         CSI$1   ,R1,R2          ; CHECK SYNTAX OF FILENAME
 000340 103521                                  BCS     110$            ; BAD FILENAME SYNTAX
 000342                                         CSI$2   ,OUTPUT         ; GET OUTPUT FILENAME
 000352 103514                                  BCS     110$            ; BAD FILENAME SYNTAX

 000354 012700  000000G                         MOV     #SYSFDB,R0      ; POINT AT SYSTEM IMAGE FILE FDB
 000360                                         OPEN$M                  ; OPEN IT
 000372 103512                                  BCS     122$            ; OPEN FAILURE

                                        ; FOOL AROUND WITH LABEL BLOCK

 000374 132767  000001  000000G                 BITB    #1,SYSFDB+F.RTYP ; SHOULD BE FIXED LENGTH RECORDS
 000402 001514                                  BEQ     130$            ; NO ??
 000404 016767  000000G 000000G                 MOV     $RSIZE,$RSIZ    ; SAVE ORIGINAL SIZE
 000412                                         FDAT$R  ,#R.FIX         ; USE FIXED LENGTH RECORDS
```

```
    787                                     .WORD   0                                                     ;[MP01]
    788                             SECFR:  .WORD   PDVTA,0         ; SECONDARY POOL BLOCKS FREE          ;[MP01]
    789                                     .RAD50  "$SECFR"                                              ;[MP01]
    790                                     .WORD   0                                                     ;[MP01]
    791                             PDVTA:  .WORD   PUMRP,0         ; PDV TABLE                           ;[MP01]
    792                                     .RAD50  "$PDVTA"                                              ;[MP01]
    793                                     .WORD   0                                                     ;[MP01]
    794                             PUMRP:  .WORD   CTLST,0         ; UMR BIAS VALUE                      ;[MP01]
    795                                     .RAD50  "$PUMR"                                               ;[MP01]
    796                                     .WORD   0                                                     ;[MP01]
    797                             CTLST:  .WORD   ICAVL,0         ; CTB LIST HEAD                       ;[MP01]
    798                                     .RAD50  "$CTLST"        ;                                     ;[MP01]
    799                                     .WORD   0                                                     ;[MP01]
    800                             ICAVL:  .WORD   CRAVL,0         ; ICB LIST                            ;[MP01]
    801                                     .RAD50  "$ICAVL"                                              ;[MP01]
    802                                     .WORD   0                                                     ;[MP01]
    803                                     .ENDC
    804
    805 003310  003322' 000000      CRAVL:  .WORD   .C00,0          ; DYNAMIC STORAGE REGION LISTHEAD
    806 003314  124512  004674              .RAD50  "$CRAVL"
    807 003320  000000                      .WORD   0
    808 003322  003334' 000000      .C00:   .WORD   DEVHD,0         ; CONSOLE OUTPUT UCB ADDRESS
    809 003326  127607  135600              .RAD50  ".C00"
    810 003332  000000                      .WORD   0
    811 003334  003346' 000000      DEVHD:  .WORD   EXSIZ,0         ; DEVICE TABLE LISTHEAD
    812 003340  124545  105304              .RAD50  "$DEVHD"
    813 003344  000000                      .WORD   0
    814 003346  003360' 000000      EXSIZ:  .WORD   FMASK,0         ; HIGHEST ADDRESS IN SYSTEM
    815 003352  124640  074102              .RAD50  "$EXSIZ"
    816 003356  000000                      .WORD   0
    817 003360  003372' 000000      FMASK:  .WORD   PARHD,0         ; SYSTEM FEATURE MASK
    818 003364  124675  004503              .RAD50  "$FMASK"
    819 003370  000000                      .WORD   0
    820 003372  003404' 000000      PARHD:  .WORD   SYSIZ,0         ; PARTITION CONTROL BLOCK LISTHEAD
    821 003376  125501  070704              .RAD50  "$PARHD"
    822 003402  000000                      .WORD   0
    823 003404  003416' 000000      SYSIZ:  .WORD   SYSNM,0         ; SYSTEM SIZE IN 32-WORD BLOCKS
    824 003410  125721  074102              .RAD50  "$SYSIZ"
    825 003414  000000                      .WORD   0
    826 003416  003430' 000000      SYSNM:  .WORD   CLINS,0         ; SYSTEM NAME
    827 003422  125721  074375              .RAD50  "$SYSNM"
    828 003426  000000                      .WORD   0
    829 003430  003442' 000000      CLINS:  .WORD   FRKHD,0
    830 003434  124504  035203              .RAD50  "$CLINS"
    831 003440  000000                      .WORD   0
    832 003442  003454' 000000      FRKHD:  .WORD   HEADR,0
    833 003446  124702  043004              .RAD50  "$FRKHD"
    834 003452  000000                      .WORD   0
    835 003454  003466' 000000      HEADR:  .WORD   STKDP,0
    836 003460  125005  003362              .RAD50  "$HEADR"
    837 003464  000000                      .WORD   0
    838 003466  003500' 000000      STKDP:  .WORD   PLGTH,0
    839 003472  125714  042560              .RAD50  "$STKDP"
    840 003476  000000                      .WORD   0
    841 003500  003512' 000000      PLGTH:  .WORD   VCTR,0          ; LENGTH OF PCB
    842 003504  064154  027350              .RAD50  "P.LGTH"
    843 003510  000000                      .WORD   0
```

```
VINI          CREATED BY  MACRO  ON 29-JUN-35 AT 02:43      PAGE 5
MACRO CROSS REFERENCE                                       CREF    04.00

MACRO NAME          REFERENCES

CALL                  7-88      7-90      7-92      7-102     7-111     7-116     8-145     8-179     8-219     8-249
                      8-251     8-253     8-254     9-356     9-362     9-364     9-397     9-410     9-415     9-429
                      9-431     9-436     9-439     9-449     9-450     9-453     10-516    10-519    11-586    11-612
                      11-614    11-621    11-623    11-628    11-671    13-951    13-986    13-988    13-991    13-993
                      14-1020   14-1021   14-1025
CALLR               #7-80       9-366
CLOSE$              #8-134
CMOV$B              #8-168    #8-168      8-168     8-168    #8-168     8-168    #8-176     8-176     8-176    #8-177
                      8-177
CMOV$W              #8-168      8-168     8-168     8-168    #8-168     8-168     8-168    #8-176     8-176     8-176
                      8-176    #8-177     8-177     8-177
CSI$                #8-162     #8-164
CSI$SW              #8-135
CSI$1               #8-134      8-162
CSI$2               #8-134      8-164
DIR$                #8-138
ERBLK$              #7-79       7-118     8-261     8-263     8-266     8-269     8-272     9-460     9-463     10-529
                      10-531    11-670    13-1000   13-1002   13-1004   13-1007
ERMSG$              #7-79       7-117     8-260     8-262     8-264     8-265     8-267     8-268     8-270     8-271
                      9-458     9-459     9-461     9-462     10-528    10-530    11-669    13-999    13-1001   13-1003
                      13-1005   13-1006
ERRPT$              #7-79       7-116     8-249     8-251     8-253     8-254     9-449     9-450     9-453     10-516
                      10-519    11-671    13-986    13-988    13-991    13-993
ERR$                #8-168      8-168
FDAT$R              #8-134      8-176
FDOFF$              #8-134      8-137
FDOP$R              #8-168      8-168
FDRC$R              #8-134     #8-168     8-168     8-177
GCLDF$              #8-135      8-138
GCL$                #8-138
GETRC               #8-134      8-179
GETRV               #7-79      #8-135     9-397     9-439     11-586    11-612    11-614    11-621    11-623    11-628
                      13-951
HWDDF$              #7-79       7-83
LBLDF$              #7-79       7-84
LDFDB$              #8-168      8-168    #8-168     8-168    #8-176     8-176    #8-177     8-177
LDRO$               #8-162      8-162    #8-164     8-164
OPEN$               #8-168      8-168
OPEN$M              #8-134      8-168
PCBDF$              #7-79       7-85
PUTRC               #8-135      8-219
RAND                #8-179      8-179    #8-219     8-219    #9-397     9-397    #9-439     9-439    #11-586    11-586
                    #11-612     11-612   #11-614    11-614   #11-621    11-621   #11-623    11-623   #11-628    11-628
                    #13-951     13-951
RETURN                7-112     8-149     8-242     9-443     10-507    11-663    13-981    14-1026
SWTCHI              #7-79       11-625    11-661
SWTCHO              #7-79       8-220
```

```
    111                                           ;
    112 000030                                    GMSG:   QIOW$   IO.WLB,1,1,,,,<,,0>
    113         000044'                           GBUF=GMSG+Q.IOPL
    114         000046'                           GLEN=GBUF+2
    115                                           ;
    116 000060                                    SCODV:  .BLKW   1               ; Alias scope device
    117 000062                                    SCOUN:  .BLKW   1               ; Alias scope unit
    118         000062'                           EVEBEG=SCOUN                    ; Event beginning
    119
    120                                           ;
    121                                           ; Ascii work area
    122                                           ;
    123 000064                                    ASCII:  .BLKB   6
    124
    125                                           ;
    126                                           ; Temporary storage for node id
    127                                           ;
    128 000072                                    NODID:  .BLKB   6               ; Node id
    129
    130 000100  000000                            AREAN:  .WORD   0               ; Area number
    131 000102  000000                            ADDR:   .WORD   0               ; Node address
    132
    133
    134
    135
    136
    137                                           ;
    138                                           ; Error messages
    139                                           ;
    140 000104     076       076                  MARKER: .ASCII  />>/
    141         000002                            LMARKER=.-MARKER
    142
    143 000106     015       012       040        CRLF:   .ASCII  <15><12>"    "
        000111     040       040       040
    144         000006                            LCRLF=  .-CRLF
    145
    146 000114                                            ETEXT$  SYNERR, <Syntax error>
    147 000137                                            ETEXT$  ILLNAD, <Illegal node address>
    148 000172                                            ETEXT$  PNAMTL, <Process name too long>
    149 000226                                            ETEXT$  LNAMTL, <Line name too long>
    150 000257                                            ETEXT$  MXLN64, <Maximum lines limited to 64>
    151 000321                                            ETEXT$  PRNMTL, <Partition name too long>
    152 000357                                            ETEXT$  CNUMTL, <Controller number too large>
    153 000421                                            ETEXT$  UNUMTL, <Unit number too large>
    154 000455                                            ETEXT$  ILLCSR, <Illegal CSR address>
    155 000507                                            ETEXT$  ILLPRI, <Illegal interrupt priority>
    156 000550                                            ETEXT$  ILLVEC, <Illegal interrupt vector>
    157 000607                                            ETEXT$  ILLCOS, <Illegal line cost>
    158 000637                                            ETEXT$  PASWTL, <Password too long>
    159 000667                                            ETEXT$  MXCN64, <Maximum controllers limited to 64>
    160 000737                                            ETEXT$  TRIBTL, <Tributary number too large>
    161 001000                                            ETEXT$  ILLUCS, <Illegal unit CSR address>
    162 001037                                            ETEXT$  ILLMAC, <Illegal multipoint active ratio>
    163 001105                                            ETEXT$  ILLMDE, <Illegal multipoint dead ratio>
    164 001151                                            ETEXT$  ONAMTL, <Owner process name too long>
    165 001213                                            ETEXT$  ILLTRI, <Illegal tributary address>
    166 001253                                            ETEXT$  ILLOBJ, <Illegal object type>
```

```
 003026  010067  000100'                 MOV     R0,AREAN        ; store area
 003032  016767  000000G 000102'         MOV     .PNUMB,ADDRS    ; store node address

 003040                                  RETURN

 003042                         101$:    ERROR$  ILLARE          ; Illegal area number


                                ;
                                ; Node address
                                ;
 003062  005767  000100'        $NADD::  TST     AREAN           ; Use default area number
 003066  001024                          BNE     10$             ; If no - branch

 003070                                  SWTCHI  #$HMBUF         ; Set up input buffer
 003076                                  GETRV   $DECPT,#D$END   ; Read decnet home block
 003116                                  SWTCHI                  ; Reset default buffer

 003124  016700  000014G                 MOV     $HMBUF+D$LNUM,R0 ; Get executor address
 003130  042700  000000C                 BIC     #^CARAMSK,R0    ; Get executor area
 003134  010067  000100'                 MOV     R0,AREAN        ; Store default area number
 003140  005767  000102'        10$:     TST     ADDRS           ; Check lower address limit
 003144  001413                          BEQ     101$            ; If illegal - branch
 003146  022767  001777  000102'         CMP     #MXADDR,ADDRS   ; Check high address limit
 003154  002407                          BLT     101$            ; If illegal - branch

 003156  016767  000102' 000072'         MOV     ADDRS,NODID     ; Store node address
 003164  056767  000100' 000072'         BIS     AREAN,NODID     ; Store area number

 003172                                  RETURN

 003174                         101$:    ERROR$  ILLNAD          ; Illegal node address


                                ;
                                ; Object task name
                                ;
 003214                         $ONAM::
 003214  012702  000006                  MOV     #6,R2           ; Set max field length
 003220                                  CALL    RAD             ; Convert task name to rad50
 003224  103405                          BCS     101$            ; If cs, more than 6 characters
 003226  010067  000004G                 MOV     R0,RQB+LO.ONM   ; Else, store object task name
 003232  010167  000006G                 MOV     R1,RQB+LO.ONM+2 ;       ...
 003236                                  RETURN
 003240                         101$:    ERROR$  OTSKTL


                                ;
                                ; Object type id
                                ;
 003260                         $OTYP::
 003260  005767  000000G                 TST     .PNUMH          ; Must be a word value
 003264  001021                          BNE     101$            ; If ne, illegal object type
 003266  105767  000001G                 TSTB    .PNUMB+1        ; Must be a byte value
 003272  001016                          BNE     101$            ; If ne, illegal value
 003274  016767  000000G 000002G         MOV     .PNUMB,RQB+LR.TYP ; Save object type
 003302  012767  000001  000010G         MOV     #1,RQB+LO.COP   ; Assume 1 copy
 003310  012767  000002  000012G         MOV     #2,RQB+LO.OFL   ; Assume verification level 2, no flags
 003316  005067  000004G                 CLR     RQB+LO.ONM      ; Assume object type zero
 003322  005067  000004G                 CLR     RQB+LO.ONM      ;       ...
```

```
                                              .SBTTL  GETLIN - GET SLT AND PDV ADDRESS FOR LINE

                                      ;
                                      ; Getlin - get SLT and PDV address for line
                                      ;
                                      ; Inputs:
                                      ;       RQB+LR.LIN = line name
                                      ;
                                      ; Outputs:
                                      ;       Carry clear:
                                      ;         R1 = PDV address
                                      ;         R2 = SLT address
                                      ;         R0 destroyed
                                      ;
                                      ;       Carry set:
                                      ;         R0,R1,R2 destroyed  - line not in system
                                      ;
006314  016700  000000G               GETLIN: MOV     $SLTMA,R0               ; Get first SLT address
006320  016746  000000G                       MOV     $SLTNM,-(SP)            ; Get number of SLT's
006324  000261                        10$:    SEC                             ; Assume error
006326  005316                                DEC     (SP)                    ; Decrement number of SLT's
006330  002412                                BLT     20$                     ; Br if no more
006332  012002                                MOV     (R0)+,R2                ; Get SLT address
006334  116201  000000G                       MOVB    L.DDM(R2),R1            ; Get ddm PDV index
006340  066701  000000G                       ADD     $PDVTA,R1               ; Point to table of PDV's
006344  011101                                MOV     (R1),R1                 ; Get PDV address
006346  026167  000000G 000002G               CMP     Z.NAM(R1),LR.LIN+RQB    ; Is this the line we want?
006354  001363                                BNE     10$                     ; Br if no
006356  006100                        20$:    ROL     R0                      ; Save carry
006360  005726                                TST     (SP)+                   ; Clean up stack
006362  006000                                ROR     R0                      ; Restore carry
006364                                        RETURN

```

```
    726                                                     .SBTTL  FNDHSH - FIND HASH TABLE ENTRY
    727
    728                                             ;+
    729                                             ;
    730                                             ; FNDHSH - FIND HASH TABLE ENTRY
    731                                             ;
    732                                             ; INPUTS:
    733                                             ;       PVCLCN - LOGICAL CHANNEL NUMBER
    734                                             ;       PVCPRT - PORT NUMBER
    735                                             ;       HSHADD - APR6 ADDRESS OF HASH TABLE ENTRY
    736                                             ;
    737                                             ; OUTPUTS:
    738                                             ;       CARRY CLEAR - TABLE ENTRY AVAILABLE AND SET UP WITH LCN/PORT
    739                                             ;       CARRY SET - NO ENTRIES AVAILABLE
    740                                             ;       R0,R1,R2 DESTROYED
    741                                             ;       MAPPING PRESERVED
    742                                             ;
    743                                             ;-
    744 002120                                      FNDHSH: SAVMAP                          ; SAVE MAPPING
    745 002124  016746  000000G                             MOV     DTEDES,-(SP)            ; GET UNMAPPED DTE DESCRIPTOR
    746 002130                                              CALL    $CEACC                  ; MAP TO LOCAL DTE (AND ALSO HASH TABLE)
    747 002134  005726                                      TST     (SP)+                   ; CLEAN STACK
    748 002136  005002                                      CLR     R2                      ; INITIAL OFFSET FOR HASH TABLE INSERTION
    749 002140  016700  000004'                             MOV     PVCLCN,R0               ; GET LOGICAL CHANNEL NUMBER
    750 002144  016701  000000G                     10$:    MOV     HSHSZ,R1                ; GET NUMBER OF ENTRIES IN HASH TABLE
    751 002150  005401                                      NEG     R1                      ; GET MASK FOR HIGH BITS
    752 002152  040100                                      BIC     R1,R0                   ; COMPUTE ENTRY NUMBER (MODULO TABLE SIZE)
    753 002154  010001                                      MOV     R0,R1                   ; SAVE ENTRY NUMBER
    754 002156  006300                                      ASL     R0                      ; MULTIPLY BY 4 TO DETERMINE OFFSET
    755 002160  006300                                      ASL     R0                      ; ... INTO HASH TABLE
    756 002162  066700  000000G                             ADD     HSHADD,R0               ; DETERMINE ENTRY ADDRESS IN HASH TABLE
    757 002166                                              GETAD   R0,#4                   ; READ IN  HASH TABLE ENTRY
    758 002204  005767  000002G                             TST     $BFR+2                  ; IS THIS ENTRY AVAILABLE?
    759 002210  001407                                      BEQ     20$                     ; BR IF YES (CLEARS CARRY)
    760 002212  005202                                      INC     R2                      ; ELSE, COLLISION - UPDATE OFFSET
    761 002214  026702  000000G                             CMP     HSHSZ,R2                ; ANY ENTRIES AVAILABLE?
    762 002220  103413                                      BLO     30$                     ; BR IF NO (SETS CARRY)
    763 002222  060201                                      ADD     R2,R1                   ; CALCULATE ENTRY NUMBER + OFFSET
    764 002224  010100                                      MOV     R1,R0                   ; SAVE ENTRY NUMBER + OFFSET
    765 002226  000746                                      BR      10$                     ; FIND NEXT AVAILABLE ENTRY
    766 002230  016767  000004' 000000G     20$:    MOV     PVCLCN,$BFR             ; STORE LOGICAL CHANNEL NUMBER IN HASH TABLE
    767 002236  116767  000024' 000002G             MOVB    PVCPRT,$BFR+2           ; STORE PORT NUMBER IN LOW BYTE OF HASH TABLE
    768 002244                                              PUTAD                           ; WRITE UPDATED HASH TABLE ENTRY
    769 002250                                      30$:    RESMAP                          ; RESTORE MAPPING
    770 002254                                              RETURN
```

```
                                        .SBTTL  LOCAL SYMBOL DEFINITIONS

                                ;
                                ; LOCAL SYMBOL DEFINITIONS
                                ;
          000006                        RDTEMX = 6              ; MAXIMUM LENGTH OF REMOTE DTE NAME
          000040                        SPACE  = 40             ; ASCII SPACE CHARACTER
```

```
VFPRNW - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 02:40
Table of content
```

```
     5-   50    Local Symbols
     6-   66    Macro references and definitions
     7-   85    Error messages
     8-   98    Local Data
     9-  107    $QRNW/$QRNA - Look for RNW$DF/RNA$DF macro
    10-  142    Parse table for RNW$DF
    11-  167    Parse table for RNA$DF
    14-  255    Syntax action routines
    19-  379    End of macros - check and allocate network block
    20-  400    CKNET   Check network name is unique
    21-  435    ALNET   Allocate remote network block
    22-  495    Parser Subexpression Action Routines
```

```
                                        ;+
                                        ; *** RNAUSR - Check username
                                        ;
                                        ; INPUTS
                                        ;       STRBUF - Contains parsed string (minus delimiters etc)
                                        ;       STRLEN - Number of characters in string
                                        ;
                                        ; OUTPUTS
                                        ;       Username checked and stored
                                        ;       Error condition if invalid username
                                        ;-

000206                                  RNAUSR:
000206  116701  000164'                         MOVB    STRLEN,R1               ; Get length of string
000212  020127  000020                          CMP     R1,#MXACUS              ; Check if username too long
000216  101011                                  BHI     101$                    ; Branch if error
000220  110167  000000G                         MOVB    R1,RNUL                 ; Move username length to data area
000224  012700  000165'                         MOV     #STRBUF,R0              ; Move username to data area
000230  012702  000000G                         MOV     #RNUSER,R2
000234                                          CALL    MOVE
000240                                          RETURN

000242                                  101$:   EMSG$R  RC                      ; Illegal access data
```

VFPRN

SYMBO

SYMBO

.PSTP
.TPAR

```
VFPRNW     CREATED BY  MACRO  ON 29-JUN-85 A1 02:40     PAGE 3
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

.PSTPT   = ******   GX      14-274      15-301
.TPARS   = ******   GX       9-131
```

M 6

```
VFPX29     CREATED BY  MACRO  ON 29-JUN-85 AT 02:41      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL   VALUE              REFERENCES

$SUBXP   = 000010            #8-124
$$$FLG   = 177777            #8-124
$$$KEY   = 177777            #8-124
.PNUMB   = ******   GX        9-158       9-160       9-170
.PNUMH   = ******   GX        9-156       9-168
.TPARS   = ******   GX        7-109
```

```
    317 000546  060167  000036G                  ADD     R1,L$MCHN+$BFR  ; POINT TO MAX CHANNELS CELL
    318 000552                                   PUTAD                   ; REWRITE DTE DESCRIPTOR
    319 000556                                   RESMAP                  ; RESTORE PREVIOUS MAPPING
    320 000562                           35$:    RETURN
    321
    322                                  ;
    323                                  ; ERRORS
    324                                  ;
    325 000564                           101$:   EMSG$R  09              ; ILLEGAL CHANNEL RANGE VALUE
    326 000572                           102$:   EMSG$R  08              ; RESOURCE ALLOCATION FAILURE
    327                                          .DSABL  LSB
    328
    329         000001                           .END
```

M 8

```
VHELP - HELP ROUTINE FOR VNP    MACRO V05.03b Saturday 29-Jun-85 02:42
Table of contents

     6-   71    $VHELP - PROCESS HELP COMMAND
```

```

                                    ;+
                                    ;       **$VIMI - READ IMAGES INTO VRES BUFFER FROM SYSTEM IMAGE
                                    ;
                                    ;
                                    ; INPUTS:
                                    ;       COMEXEC MUST BE LOADED
                                    ;       R2=TABLE ADDRESSES RELOCATION BIAS
                                    ;       .PDVTA  - ADDRESS OF FIRST WORD OF CETAB IN SYSTEM IMAGE
                                    ;       .SLTMB  - ADDRESS OF SYSTEM LINE MAPPING TABLE IN SYSTEM IMAGE
                                    ;       $PDVTA  - LABEL ON FIRST WORD OF CETAB TEMPLATE IN VRES
                                    ;       $NTLPT  - CONTAINS BIAS OF FIRST BLOCK OF NETWORK POOL
                                    ;               - IF UNMAPPED, IT CONTAINS THE ADDRESS
                                    ;
                                    ; OUTPUTS:
                                    ;       ALWAYS SUCCESSFUL ( ERROR EXIT TAKEN THROUGH $PANIC )
                                    ;
                                    ;       TEMPLATE STARTING AT $PDVTA FILLED IN SO THAT ALL REFERENCES
                                    ;       TO CETAB CAN BE MADE EITHER DIRECTLY, OR INDIRECTLY THROUGH
                                    ;       $PDVTA,$LLCTA,&$SLTMA.
                                    ;
                                    ;       VNP VIRTUAL I/O ROUTINES WILL BE USING THE DEFAULT BUFFER $BFR
                                    ;
                                    ;       ALL REGISTERS ARE PRESERVED
                                    ;-
                                            .MCALL  GETRV,PUTRC,SWTCHI,GETAD,ERMSG$,ERBLK$,ERRPT$,HWDDF$    ; RG002
                                                                                                            ; RG002
000000                                      HWDDF$                          ; DEFINE THE HARDWARE REGISTERS ; RG002
                                                                            ; AND FEATURES MASK BITS        ; RG002

                                            .IF DF  R$$MPL                                                  ;[MP01]
                                            .MCALL  CTBDF$                                                  ;[MP01]
                                            CTBDF$                                                          ;[MP01]
                                            .ENDC                                                           ;[MP01]

                                            .MCALL  PHBDF$                                                          ;[TD001]
000000                                      PHBDF$                          ; Define the PSI homeblock flags        ;[TD001]

                                            .MCALL  VNPDBG                                                          ;[TD001]
                                                                                                                    ;[TD001]
                                            .ENABLE LC

000000                              $VIMI:: VNPDBG  VIMI            ; VNP DEBUGGING BREAK POINT
000000                                      CALL    $SAVAL          ; SAVE ALL REGISTERS
000004                                      SWTCHI  #$PDVTA         ; POINT AT CEXCM TEMPLATE IN VRES
000012  012767  000000C 000000G             MOV     #<$CEXND-$PDVTA>,$RSIZE ; STORE SIZE OF CEXCM
000020                                      GETRV   .PDVTA          ; READ IN CEXCM

000032                                      SWTCHI  #$CETAB         ; POINT AT CETAB TEMPLATE IN VRES
000040  012704  000000C                     MOV     #<$CETND-$CETAB>,R4 ; GET SIZE OF CET$$A PSECT
000044  010467  000000G                     MOV     R4,$RSIZE       ; ASSUME LE 512. BYTES
000050  022704  001000                      CMP     #512.,R4        ; IS IT TRUE ?
000054  002003                              BGE     5$              ; IF GE, YES
000056  012767  001000  000000G             MOV     #512.,$RSIZE    ; LIMIT TRANSFER TO 512. BYTES
                                    ;                                                               ; RG002
                                    ; REDUCE $RSIZE IF NECESSARY                                    ; RG002
```

```
                                        ;+
                                                **$VIMO - CETAB IS UPDATED IN SYS IMG FROM VRES TEMPLATE
                                        ;
                                        ; INPUTS:
                                        ;       TEMPLATE STARTING AT $PDVTA
                                        ;       .PDVTA  - ADDRESS OF CETAB IN SYSTEM IMAGE
                                        ;       .SLTMB  - ADDRESS OF SYSTEM LINE MAPPING TABLE IN SYSTEM IMAGE
                                        ;
                                        ; OUTPUTS:
                                        ;       ALWAYS SUCCESSFUL, ( ERRORS EXIT THROUGH $PANIC )
                                        ;       VNP VIRTUAL I/O ROUTINES WILL BE USING THE DEFAULT BUFFER $BFR
                                        ;-

                                                .MCALL  GETRV,PUTRC,SWTCHO,PUTAD
                                                .ENABL  LC

 000000                                 $VIMO::
                                                .MCALL  VNPBPT
 000000                                         VNPBPT  <VIMO>
 000000                                         CALL    $SAVAL          ; SAVE ALL REGISTERS
                                        ;
                                        ; RELOCATE ADDRESSES
                                        ;
 000004  016700  000000G                        MOV     $CETDF,R0       ; GET SHIFT DIFFERENTIAL
 000010  016701  000000G                        MOV     $PDVTA,R1       ; GET ADDRESS OF PDV TABLE
 000014  016702  000000G                        MOV     $PDVNM,R2       ; AND NUMBER OF PDV ENTRIES
 000020  001403                                 BEQ     4$              ; IF EQ, NONE ???
 000022  060021                         3$:     ADD     R0,(R1)+        ; RELOCATE PDV ADDRESSES
 000024  005302                                 DEC     R2              ; ANY MORE ?
 000026  003375                                 BGT     3$              ; IF GT YES

 000030  016701  000000G                4$:     MOV     $SLTMA,R1       ; GET ADDRESS OF SLT TABLE
 000034  016702  000000G                        MOV     $SLTNM,R2       ; AND NUMBER OF SLT ENTRIES
 000040  001403                                 BEQ     6$              ; IF EQ, NONE ???
 000042  060021                         5$:     ADD     R0,(R1)+        ; RELOCATE SLT ADDRESSES
 000044  005302                                 DEC     R2              ; ANY MORE ?
 000046  003375                                 BGT     5$              ; IF GT, YES

 000050  016702  000000G                6$:     MOV     $LLCTA,R2       ; GET ADDRESS OR REVERSE MAPPING TABLE
 000054  016701  000000G                        MOV     $SLTNM,R1       ; AND NUMBER OF ENTRIES
 000060  001412                                 BEQ     9$              ; IF EQ, NONE
 000062  010003                                 MOV     R0,R3           ; COPY SHIFT DIFFERENTIAL
 000064  006203                                 ASR     R3              ; ADJUST SHIFT DIFFERENTIAL
 000066  005712                         7$:     TST     (R2)            ; IS THIS A MULTIPOINT LINE ?
 000070  100003                                 BPL     8$              ; IF PL, NO
 000072  060312                                 ADD     R3,(R2)         ; ELSE RELOCATE ADDRESS
 000074  052712  100000                         BIS     #100000,(R2)    ; MAKE SURE SIGN BIT IS SET !
 000100  005722                         8$:     TST     (R2)+           ; SKIP ENTRY
 000102  005301                                 DEC     R1              ; ANY MORE ?
 000104  003370                                 BGT     7$              ; IF GT, YES
                                        ;
                                        ; RELOCATE TABLE POINTERS
                                        ;
 000106  060067  000000G                9$:     ADD     R0,$PDVTA       ; RELOCATE PDV TABLE POINTER
 000112  060067  000000G                        ADD     R0,$SLTMA       ; RELOCATE SLT TABLE POINTER
 000116  060067  000000G                        ADD     R0,$LLCTA       ; RELOCATE LLC TABLE POINTER
                                        ;
```

```
000420                                    FDRC$R  ,#FD.INS.,#512. ; DON'T TRUNCATE FILE ON PUT$

000434                                    GETRC   #-<2*512.>,#L$BBLK+2    ; READ LABEL BLOCK
000454  026727  000360G 000002            CMP     $BFR+L$BBLK,#2  ; IS THIS REALLY A SYSTEM IMAGE ?
000462  001064                            BNE     130$            ; CONTEXT ERROR
000464  022767  177777  000350G           CMP     #-1,$BFR+L$BXFR ; CORRUPT SYSTEM IMAGE?
000472  001471                            BEQ     132$            ; BR IF YES
000474  016746  000350G                   MOV     $BFR+L$BXFR,-(SP) ; SAVE TRANSFER ADDRESS

000500  021627  000060                    CMP     (SP),#60        ; IS THIS A VIRGIN SYSTEM IMAGE?
000504  101014                            BHI     15$             ; BR IF YES


                                          .IF DF  R$$MPL                                          ;[MP01]
                                          MOV     (SP),$XFRSV     ; SAVE XFER ADDRESS             ;[MP01]
                                          .ENDC                                                   ;[MP01]

000506  005067  000000G                   CLR     $LOWSZ          ; INITIALIZE NUMBER OF BYTES SAVE
                                                                  ;    DISPLACED FROM LOW MEMORY
000512  005216                            INC     (SP)            ; FORM NUMBER OF BLOCKS IN BOOT BLOCK
000514  062767  001000  000000G 12$:      ADD     #1000,$LOWSZ    ; FORM NUMBER OF BYTES IN BOOT BLOCK
000522  005316                            DEC     (SP)            ; ...
000524  001373                            BNE     12$             ; ...

                                          .IF NDF R$$MPL                                          ;[MP01]
000526  062767  000200  000000G           ADD     #200,$LOWSZ     ; PLUS 200 FOR M SAVE           ;[MP01]
                                          .IFF                                                    ;[MP01]
                                          ADD     #24,$LOWSZ      ; PLUS 24 BYTES DISPLACED BY M+ SAV ;[MP01]
                                          .ENDC                                                   ;[MP01]
000534  000400                            BR      16$             ; CONTINUE                      ;[MP01]

000536                          15$:
                                          .IF DF  R$$MPL                                          ;[MP01]
                                          MOV     (SP)+,$XFRSV    ; SAVE XFER ADDRESS             ;[MP01]
                                          ERRPT$  ERRO,$EROUT     ; SYSTEM NOT SAVED              ;[MP01]
                                          SEC                     ; INDICATE ERROR

                                          RETURN
                                          .ENDC                                                   ;[MP01]

000536                          16$:
000536  012667  000000G                   MOV     (SP)+,$XFRAD    ; TRANSFER ADDRESS (0 IF SAVED SYSTEM)
000542  012767  177777  000350G           MOV     #-1,$BFR+L$BXFR ; SET CORRUPT FLAG ( REMOVED IN VXIT )
000550                                    PUTRC                   ; REWRITE LABEL BLOCK
000554                                    SWTCHO                  ; INITIALIZE OUTPUT BUFFER POINTER

                                          .IF NDF R$$MPL                                          ;[MP01]

000562  012700  000000G                   MOV     #FILE,R0        ; TRUNCATE FILENAME STRING AT /SS
000566  122710  000057          20$:      CMPB    #'/,(R0)        ;  IF IT EXISTS ...
000572  001402                            BEQ     30$             ; ...
000574  105720                            TSTB    (R0)+           ; EOS ?
000576  001373                            BNE     20$             ; NOT YET
000600  105010                  30$:      CLRB    (R0)            ; TERMINATE STRING ( MAYBE TWICE )

                                          .IFF                                                    ;[MP01]

                                          SWTCHI  #$VCTVB         ; SET UP INPUT ADDRESS          ;[MP01]
```

```
003512  003524' 000000          VCTR:   .WORD   NS0,0           ; HIGHEST LEGAL INTERRUPT ADDRESS + 4
003516  106723  012762                  .RAD50  "V$$CTR"
003522  000000                          .WORD   0
003524  003536' 000000          NS0:    .WORD   NS1,0           ; ERROR LOG INTERRUPT ( VECTORS 0 - 76 )
003530  125403  135600                  .RAD50  "$NS0"
003534  000000                          .WORD   0
003536  003550' 000000          NS1:    .WORD   NS2,0           ; VECTORS 100 - 176
003542  125403  140700                  .RAD50  "$NS1"
003546  000000                          .WORD   0
003550  003562' 000000          NS2:    .WORD   NS3,0           ; VECTORS 200 - 276
003554  125403  144000                  .RAD50  "$NS2"
003560  000000                          .WORD   0
003562  003574' 000000          NS3:    .WORD   NS4,0           ; VECTORS 300 - 376
003566  125403  147100                  .RAD50  "$NS3"
003572  000000                          .WORD   0
003574  003606' 000000          NS4:    .WORD   NS5,0           ; VECTORS 400 - 476
003600  125403  152200                  .RAD50  "$NS4"
003604  000000                          .WORD   0
003606  003620' 000000          NS5:    .WORD   NS6,0           ; VECTORS 500 - 576
003612  125403  155300                  .RAD50  "$NS5"
003616  000000                          .WORD   0
003620  003632' 000000          NS6:    .WORD   NS7,0           ; VECTORS 600 - 676
003624  125403  160400                  .RAD50  "$NS6"
003630  000000                          .WORD   0
003632  003644' 000000          NS7:    .WORD   MGEXT,0         ; VECTORS 700 - 776
003636  125403  163500                  .RAD50  "$NS7"
003642  000000                          .WORD   0
003644  003656' 000000          MGEXT:  .WORD   MGMGE,0         ; 11/70 EXTENDED MEMORY SUPPORT
003650  052623  021424                  .RAD50  "M$$EXT"
003654  000000                          .WORD   0
003656  003670' 000000          MGMGE:  .WORD   UMRHD,0         ; MEMORY MANAGEMENT
003662  052623  051135                  .RAD50  "M$$MGE"
003666  000000                          .WORD   0
003670  003702' 000000          UMRHD:  .WORD   DEUMR,0         ; FREE UMR LISTHEAD
003674  126025  070704                  .RAD50  "$UMRHD"
003700  000000                          .WORD   0
003702  003714' 000000          DEUMR:  .WORD   TSKHD,0         ; EXEC UMR ROUTINE
003706  124545  102532                  .RAD50  "$DEUMR"
003712  000000                          .WORD   0
003714  003726' 000000          TSKHD:  .WORD   ABTIM,0         ; TCB LISTHEAD
003720  125763  043004                  .RAD50  "$TSKHD"
003724  000000                          .WORD   0
003726  003740' 000000          ABTIM:  .WORD   TLGTH,0         ; ABSOLUTE TIME COUNTER
003732  124352  077165                  .RAD50  "$ABTIM"
003736  000000                          .WORD   0
003740  003752' 000000          TLGTH:  .WORD   CLKHD,0         ; TCB LENGTH
003744  100554  027350                  .RAD50  "T.LGTH"
003750  000000                          .WORD   0
003752  003764' 000000          CLKHD:  .WORD   PATT,0          ; CLOCK QUEUE LISTHEAD
003756  124504  043004                  .RAD50  "$CLKHD"
003762  000000                          .WORD   0
003764  003776' 000000          PATT:   .WORD   PHDR,0          ; PCB ATTACH LISTHEAD
003770  064141  100040                  .RAD50  "P.ATT"
003774  000000                          .WORD   0
003776  004010' 000000          PHDR:   .WORD   PPRO,0          ; PCB TASK HEADER OFFSET
004002  064150  015720                  .RAD50  "P.HDR"
004006  000000                          .WORD   0
```

```
**FILE**ID**VLBIO




VV      VV    LL           BBBBBBBB        IIIIII       000000
VV      VV    LL           BBBBBBBB        IIIIII       000000
VV      VV    LL           BB       BB       II       00      00
VV      VV    LL           BB       BB       II       00      00
VV      VV    LL           BB       BB       II       00      00
VV      VV    LL           BB       BB       II       00      00
VV      VV    LL           BBBBBBBB          II       00      00
VV      VV    LL           BBBBBBBB          II       00      00
VV      VV    LL           BB       BB       II       00      00
VV      VV    LL           BB       BB       II       00      00
  VV  VV      LL           BB       BB       II       00      00        ....
  VV  VV      LL           BB       BB       II       00      00        ....
    VV        LLLLLLLLLL   BBBBBBBB        IIIIII       000000          ....
    VV        LLLLLLLLLL   BBBBBBBB        IIIIII       000000          ....




LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
    167 001305                                      ETEXT$  OCOPTL, <Too many object task copies>
    168 001347                                      ETEXT$  OTSKTL, <Object task name too long>
    169 001407                                      ETEXT$  IVLIID, <Invalid line identification>
    170 001451                                      ETEXT$  GETFAL, <Failed to get command line>
    171 001512                                      ETEXT$  ACCTTL, <Account field too long>
    172 001547                                      ETEXT$  UIDTL,  <User id field too long>
    173 001604                                      ETEXT$  PSWFTL, <Password field too long>
    174 001642                                      ETEXT$  NSTERM, <Terminal not in system>
    175 001677                                      ETEXT$  MSUNUM, <Missing parameter, unit number>
    176 001744                                      ETEXT$  MSTRIB, <Missing parameter, tributary number>
    177 002016                                      ETEXT$  NSLINE, <Line not in system>
    178 002047                                      ETEXT$  BDGRP,  <Invalid parameter grouping>
    179 002110                                      ETEXT$  BDHELT, <Bad Hello-timer value>
    180 002144                                      ETEXT$  BDLIST, <Bad Listen-timer value>
    181 002201                                      ETEXT$  NAMETL, <Name too long>
    182 002225                                      ETEXT$  IVDTEA, <Invalid DTE address>
    183 002257                                      ETEXT$  DTEATL, <DTE address too long>
    184 002312                                      ETEXT$  ILLARE, <Illegal area number>
    185 002344                                      ETEXT$  ILLSEG, <Illegal NSP segment size>
    186 002403                                      ETEXT$  ILLINC, <Illegal incoming timer value>
    187 002446                                      ETEXT$  ILLRET, <Illegal retransmit factor value>
    188 002514                                      ETEXT$  ILLINA, <Illegal inactivity timer value>
    189 002561                                      ETEXT$  ILLOUT, <Illegal outgoing timer value>
    190
    191                                             .EVEN
    192 000000                                      .PSECT
```

```
003326                                            RETURN
003330                                  101$:     ERROR$   ILLOBJ

                                        ;
                                        ; Object verification level specified
                                        ;
003350                                  $OVER::
003350  042767  000007  000012G                   BIC      #7,RQB+LO.OFL      ; Clear out the verification level
003356                                            RETURN

                                        ;
                                        ; Line owner
                                        ;
003360  012702  000003                  $OWN::    MOV      #3,R2              ; Set max field length
003364                                            CALL     RAD                ; Get 3 RAD50 characters
003370  103403                                    BCS      101$               ; If CS, more than 3
003372  010067  000026G                           MOV      P0,RQB+LO.OWN      ; Else, store line owner
003376                                            RETURN
003400                                  101$:     ERROR$   ONAMTL

                                        ;
                                        ; Store PSI circuit name
                                        ;
003420                                  $PCIR::
003420  026727  000000G 000000G                   CMP      $CMAND,#CM$SHO     ; If not show command, do not accept name          ;[TD001]
003426  001010                                    BNE      11$                                                                   ;[TD001]
003430  012700  000002G                           MOV      #RQB+LR.LIN,R0     ; Buffer addr                                      ;[TD001]
003434  012701  000006                            MOV      #MXPSCN,R1         ; Max. size of old-format circuit name             ;[TD001]
003440                                            CALL     STORID             ; Check and store circuit name                     ;[TD001]
003444  103404                                    BCS      101$               ; Branch if name too long                          ;[TD001]
003446                                            RETURN                                                                        ;[TD001]
                                                                                                                                ;[TD001]
003450  062716  000002                  11$:      ADD      #2,(SP)            ; Don't recognize the name.                        ;[TD001]
003454                                            RETURN                                                                        ;[TD001]
                                                                                                                                ;[TD001]
003456                                  101$:     ERROR$   NAMETL             ; Show syntax error                                ;[TD001]

                                        ;
                                        ; Process name
                                        ;
003476  012702  000003                  $PNAM::   MOV      #3,R2              ; Set max field length
003502                                            CALL     RAD                ; Get 3 rad50 characters
003506  103403                                    BCS      101$               ; If cs, more than 3
003510  010067  000002G                           MOV      R0,RQB+LR.PRO      ; Set process name
003514                                            RETURN
003516                                  101$:     ERROR$   PNAMTL

                                        ;
                                        ; Priority value
                                        ;
003536  005767  000000G                 $PRI::    TST      .PNUMH             ; Must be single precision
003542  001011                                    BNE      101$               ;  ...
003544  022767  000007  000000G                   CMP      #7,.PNUMB          ; And less than 8.
003552  103405                                    BLO      101$               ;  ...
003554  016767  000000G 000012G                   MOV      .PNUMB,RQB+LO.PRI  ; Set priority value
003562  001401                                    BEQ      101$               ; Zero value is illegal
003564                                            RETURN
```

```
                                              .SBTTL  CHKLOG - CHECK FOR LOGGING FUNCTION

                                     ;+
                                     ;
                                     ; Chklog - Check for logging function
                                     ;
                                     ; Inputs:
                                     ;        RQB+LR.STS - function code
                                     ;
                                     ; Outputs:
                                     ;        Carry clear:
                                     ;           Function is a logging function
                                     ;        Carry set:
                                     ;           Function is not a logging function
                                     ;
                                     ;-
 006366  122767  000020  000000G     CHKLOG: CMPB    #LS.LOG,LR.STS+RQB      ; Set logging?
 006374  001406                              BEQ     10$                     ; Br if yes
 006376  122767  000021  000000G             CMPB    #LX.LOG,LR.STS+RQB      ; Clear logging?
 006404  001402                              BEQ     10$                     ; Br if yes
 006406  000261                              SEC                             ; Else not logging function
 006410  000401                              BR      20$                     ; Finish up
 006412  000241                      10$:    CLC                             ; Indicate logging function
 006414                              20$:    RETURN
```

```
                                            .SBTTL  BIASX - BIAS XPOOL ADDRESS FOR APR6

                                    ;+
                                    ; BIASX - BIAS XPOOL ADDRESS FOR APR6
                                    ;
                                    ; INPUTS:
                                    ;       R0 = ADDRESS (POOL OR APR5)
                                    ;
                                    ; OUTPUTS:
                                    ;       R0 = ADDRESS (POOL OR APR6)
                                    ;
                                    ;-

002256                              BIASX:
002256  020027  120000                      CMP     R0,#120000      ; POOL ADDRESS?
002262  103404                              BLO     10$             ; YES, LEAVE AS IS
002264  042700  160000                      BIC     #160000,R0      ; CLEAR APR BITS
002270  052700  140000                      BIS     #140000,R0      ; BIAS FOR APR6
002274                              10$:    RETURN                  ; RETURN
```

```
     86                                          .SBTTL  LOCAL DATA
     87
     88                                          .NLIST  BEX
     89 000000                                   .PSECT  DATA,D
     90                                  ;
     91                                  ; LOCAL DATA FOR RDT$DF MACRO
     92                                  ;
     93 000000                           RDTNAM: .BLKB   6                ; REMOTE DTE NAME
     94         000017                           RDTLN = 15.
     95 000006                           RDTE:   .BLKB   RDTLN            ; REMOTE DTE ADDRESS
     96         000024                           RDTEN = .-1              ; END OF DTE ADDRESS
     97 000025                           RDTEP:  .BLKB   <RDTLN+1>/2      ; REMOTE DTE ADDRESS IN BCD
     98                                          .EVEN
```

```
        .TITLE  VFPRNW - INITIAL CONFIG FILE SCAN ACTION ROUTINES
        .IDENT  /V05.00/
        .ENABL  LC
;
; COPYRIGHT (C) 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       Parse tables and action routines for processing
;       CETAB PNW$DF and RNA$DF macros.
;
; EUROPEAN DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  11-OCT-82
;       First written for X25 Gateway RSX Host Access
;       [Tim Dixon EDSG]
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
    332                                                 ;+
    333                                                 ; *** RNAPAS - Check password
    334                                                 ;
    335                                                 ; INPUTS
    336                                                 ;       STRBUF - Contains parsed string (minus delimiters etc)
    337                                                 ;       STRLEN - Number of characters in string
    338                                                 ;
    339                                                 ; OUTPUTS
    340                                                 ;       Password checked and stored
    341                                                 ;       Error condition if invalid password
    342                                                 ;-
    343 000250                                          RNAPAS:
    344 000250  116701  000164'                                 MOVB    STRLEN,R1               ; Get length of string
    345 000254  020127  000010                                  CMP     R1,#MXACPA              ; Check if password too long
    346 000260  101011                                          BHI     101$                    ; Branch if error
    347 000262  110167  000000G                                 MOVB    R1,RNPL                 ; Move password length to data area
    348 000266  012700  000165'                                 MOV     #STRBUF,R0              ; Move password to data area
    349 000272  012702  000000G                                 MOV     #RNPASS,R2
    350 000276                                                  CALL    MOVE
    351 000302                                                  RETURN
    352
    353 000304                                          101$:   EMSG$R  RC                      ; Illegal access data
```

```
VFPRN
MACRO
MACRO
CALL
CEACC
DBGTF
EMSG$
GETAD
GETRV
ISTAT
MTRAN
NTLEF
PHBDF
PUTAD
RAND
RESMA
RETUF
RNWDF
SAVMA
SOB
STATE
TRAN
VNPD
```

```
                                                                  N  5
VFPRNW     CREATED BY  MACRO  ON 29-JUN-85 AT 02:40      PAGE 4
MACRO CROSS REFERENCE                                    CREF   04.00

MACRO NAME       REFERENCES

CALL              9-131     14-276    15-303    16-327    17-350    18-374    19-391    19-393    20-417    20-421
                 20-422     21-449    21-452    21-455    21-465    21-468    21-472    21-477    21-482    21-487
                 21-488     21-491
CEACC$           #6-69      20-421    21-455
DBGTP$           #10-144   #11-178   #11-182   #11-188   #12-210   #12-214   #13-226   #13-227   #13-231   #13-234
                 #13-235   #13-243   #13-246   #13-250   #13-251
EMSG$R           #6-68       9-139    14-283    15-306    16-330    17-353    18-377    19-397    19-398
GETAD            #6-69      20-422
GETRV            #6-69      20-417
ISTAT$           #6-68      10-144
MTRAN$           #10-144
NTLER$           #6-68       7-90      7-91      7-92      7-93
PHBDF$           #6-68       6-73
PUTAD            #6-69      21-488
RAND             #20-417    20-417   #20-422    20-422   #21-488    21-488
RESMAP           #6-69      #6-81     20-432    21-492
RETURN            9-135     14-281    15-304    16-328    17-351    18-375    19-395    20-433    21-493    22-506
                 23-524
RNWDF$           #6-68       6-72
SAVMAP           #6-69      #6-77     20-414    21-450
SOB              20-430     21-459
STATE$           #6-68      10-146   #10-148   #10-151   #10-153   #10-156   #10-159   #10-162   #10-165    11-169
                 #11-171   #11-174   #11-176   #11-180   #11-183   #11-186   #11-189   #11-192   #11-194   #12-199
                 #12-201   #12-203   #12-205   #12-208   #12-212   #13-224   #13-229   #13-232   #13-236   #13-238
                 #13-241   #13-244   #13-248   #13-253
TRAN$            #6-68     #10-147   #10-149   #10-152   #10-154   #10-157   #10-160   #10-163   #11-170   #11-172
                 #11-175   #11-177   #11-178   #11-181   #11-182   #11-184   #11-187   #11-188   #11-190   #11-193
                 #12-200   #12-202   #12-204   #12-206   #12-209   #12-210   #12-213   #12-214   #13-225   #13-226
                 #13-227   #13-230   #13-231   #13-233   #13-234   #13-235   #13-237   #13-239   #13-242   #13-243
                 #13-245   #13-246   #13-249   #13-250   #13-251
VNPDBG           #6-70       9-125
```

```
                                                                N  6
VFPX29      CREATED BY  MACRO  ON 29-JUN-85 AT 02:41      PAGE 3

MACRO CROSS REFERENCE                                     CREF    04.00

MACRO NAME       REFERENCES

CALL              7-109     9-181     9-186     9-197     9-200     9-202
DBGTP$           #8-124    #8-147
EMSG$R           #5-57      7-118     9-163     9-173     9-205     9-206
GETRV            #5-58      9-181     9-200
ISTAT$           #5-57      8-124
MTRAN$           #8-124
NTLER$           #5-57      6-74      6-75      6-76      6-77
PHBDF$           #5-57      5-61
PUTRC            #5-58      9-197     9-202
RAND             #9-181     9-181    #9-197     9-197    #9-200     9-200    #9-202     9-202
RETURN            7-113     9-161     9-171     9-203
STATE$           #5-57      8-128    #8-131    #8-134    #8-137    #8-140    #8-145    #8-149
TRAN$            #5-57     #8-129    #8-132    #8-135    #8-138    #8-141    #8-146    #8-147
X29DF$           #5-57      5-60
```

```
Symbol table

A$$CHK= 000000          G$$TTK= 000000          L$GLEN  000134          M$$FCS= 000000          $BLANK= 000006
A$$CPS= 000000          G$$WRD= 000000          L$LEN   000122          M$$MGE= 000000          $CEACC= ****** GX
A$$PRI= 000000          I$$RAR= 000000          L$LLCH  000016          M$$NET= 000000          $DIGIT= 000024
A$$TRP= 000000          I$$RDN= 000000          L$LNK   000000          M$$OVR= 000000          $DNUMB= 000014
BLKDEF  000060R         K$HIC   000004          L[illegible]CHN  000036          N$$ACC= 000001          $EOS   = 000012
BLKMAX  000102R         K$LEN   000006          L$NETW  000114          N$$BUF= 000001          $ERR00= ****** GX
BLKSZD= ****** GX       K$LNK   000000          L$NIRE  000072          N$$LDV= 000001          $ERR01= ****** GX
BLKSZM= ****** GX       K$LOC   000002          L$NLRE  000104          N$$MCP= 000001          $ERR02= ****** GX
CERR   = ****** GX      K$$CNT= 177546          L$NMAC  000076          N$$MLL= 000001          $ERR08  000004R      002
CFGBF  = ****** GX      K$$CSR= 177546          L$NMAS  000074          N$$MOV= 000010          $ERR09= ****** GX
CFGSZ  = ****** GX      K$$LDC= 000000          L$NMST  000003          N$$NCT= 000001          $ERRP7= ****** GX
CFLIN  = ****** GX      K$$TPS= 000074          L$NNRE  000106          N$$PEM= 000001          $ERRP8= ****** GX
CHEND   000302R         LD$LP  = 000000         L$NRBY  000046          P$$P45= 000000          $ERRP9= ****** GX
CHHI    000254R         LN$OFF= 000001          L$NRCA  000066          P$$WRD= 000000          $ERR1T= ****** GX
CHKBLK= ****** GX       LN$ON  = 000000         L$NRFS  000100          Q$$OPT= 000010          $EXIT  = 000000
CHKWND= ****** GX       LN$SHU= 000002          L$NRPK  000056          RTSPC  = ****** GX      $FAIL  = 177777
CHLD    000220R         LP$EIP= 002000          L$NRRE  000105          R$$DER= 000000          $GETAD= ****** GX
CHNDF   000036R     003 LP$ENB= 004000          L$NRST  000107          R$$K11= 000001          $GETRV= ****** GX
CHNLHI  000002R     002 LP$PCT= 001400          L$NTBY  000052          R$$SND= 000000          $LAMDA= 000000
CHNLLD  000000R     002 LP$TMR= 100000          L$NTCA  000070          R$$11M= 000000          $NUMBR= 000002
CHNLMN= 000001          LP$UP  = 010000         L$NTFS  000102          SYNERR= ****** GX       $PUTAD= ****** GX
CHNLMX= ****** GX       LP$WTD= 020000          L$NTPK  000062          SYSFDB= ****** GX       $PUTRC= ****** GX
C$$CKP= 000000          LP$WTS= 040000          L$OMST  000002          S$$WRG= 000000          $QCHN   000006RG
C$$ORE= 000400          LS$GDN= 000004          L$PCS   000112          S$$YSZ= 007600          $QX3P   000000RG
C$$RSH= 177564          LS$OFF= 000000          L$PLS   000110          T$$KMG= 000000          $RADDR= ****** GX
DTEDES= ****** GX       LS$STR= 000002          L$QCNT  000122          T$$MIN= 000000          $RAD50= 000016
D$$BUG= 177514          LS$UP  = 000003         L$QUEH  000124          V$$CTR= 001000          $RBLCK= ****** GX
D$$ISK= 000000          LS$WT  = 000001         L$QUET  000130          WNDDEF  000140R         $SAVAL= ****** GX
D$$L11= 000001          L$ACHN  000012          L$RTRY  000007          WNDMAX  000162R         $STRNG= 000004
D$$YNC= 000000          L$APVC  000014          L$SLN   000004          WNDSZD= ****** GX       $SUBXP= 000010
D$$YNM= 000000          L$ASVC  000010          L$ST    000005          WNDSZM= ****** GX       $$$FLG= 177777
END     000070R     003 L$AUC   000042          L$TCLZ  000044          X$$DBT= 000000          $$$KEY= 000001
E$$XPR= 000000          L$CHLS  000034          L$TIM   000006          X3PDF   000000R     003 $$$R   = 000010
FMT8   = ****** GX      L$CHTB  000030          L$$ASG= 000000          X3PKW   000000RG    004 $$$STA= 000000
FM.B   = 000000         L$CLEN= 000044          L$$DRV= 000000          X3PST   000000RG    003 $$$TMP= 000007R      005
F$$LVL= 000001          L$CTEN  000032          L$$P11= 000001          $ALL16= ****** GX        .PNUMB= ****** GX
F.RSIZ= ****** GX       L$CTIM  000040          L$$11R= 000000          $ALPHA= 000022           .PNUMH= ****** GX
G$$TPP= 000000          L$DTEA  000020          M$$CRB= 000124          $ANY   = 000020          .TPARS= ****** GX
G$$TSS= 000000          L$DTEL  000017          M$$CRX= 000060          $BFR   = ****** GX

. ABS.  000134    000   (RW,I,GBL,ABS,OVR)
        000600    001   (RW,I,LCL,REL,CON)
DATA    000056    002   (RW,D,LCL,REL,CON)
$STATE  000100    003   (RW,D,LCL,REL,CON)
$KTAB   000004    004   (RW,D,LCL,REL,CON)
$KSTR   000016    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13997 Words  ( 55 Pages)
Size of core pool: 15496 Words  ( 59 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:30.52
```

```
                                        .TITLE  VHELP - HELP ROUTINE FOR VNP
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ; VNP - HANDLE PROCESSING OF HELP INFORMATION
                                ;
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  14-DEC-79
                                ;       DECNET-11M/S V3.0
                                ;       DECNET-11M-PLUS V1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;
```

```
                                        ;                                                       ; RG002
000064  016701  000000G                 5$:     MOV     $PDVTA,R1       ; CALCULATE LARGEST ADDRESS ALLOWED     ; RG002
000070  016767  000000G 000000G                 MOV     $PDVTA,$CETST   ; SAVE ACTUAL START OF CETAB
000076  066701  000000G                         ADD     $RSIZE,R1       ; CALCUALTE END OF CETAB                ; RG002
000102  012746  160000                          MOV     #160000,-(SP)   ; ASSUME UN-MAPPED SYSTEM               ; RG002
000106  005767  000000G                         TST     .MGMGE          ; IS THIS A MAPPED SYSTEM ?             ; RG002
000112  100410                                  BMI     7$              ; IF MI, NO                             ; RG002
000114  012716  120000                          MOV     #120000,(SP)    ; ASSUME 20K EXEC                       ; RG002
000120  032767  000004  000000G                 BIT     #FE.EXV,$FMASK  ; IS THIS A 20K EXEC SYSTEM ?           ; RG002
000126  001002                                  BNE     7$              ; IF NE, YES                            ; RG002
000130  012716  100000                          MOV     #100000,(SP)    ; ELSE, THIS MUST BE A 16K EXEC         ; RG002
000134  162601                          7$:     SUB     (SP)+,R1        ; IS THE ADDRESS TOO BIG ?              ; RG002
000136  103402                                  BCS     8$              ; IF CS, NO                             ; RG002
000140  160167  000000G                         SUB     R1,$RSIZE       ; CALCULATE MAXIMUM SIZE ALLOWED        ; RG002
000144                                  8$:     GETRV   $PDVTA          ; READ IN AS MUCH AS POSSIBLE

                                        ; DETERMINE SHIFT DIFFERENTIAL AND BIAS ADDRESS POINTERS BY THAT DIFFERENTIAL

000156  016700  000000G                         MOV     $PDVTA,R0       ; ADDRESS OF FIRST WORD OF CETAB IN SYS IMG
000162  162700  000000G                         SUB     #$CETAB,R0      ; SUBTRACT IMAGE START ADDRESS
000166  010067  000000G                         MOV     R0,$CETDF       ; STORE DIFFERENCE IN ROOT SEGEMNET

000172  160067  000000G                         SUB     R0,$PDVTA       ; SUBTRACT
000176  160067  000000G                         SUB     R0,$LLCTA       ;  SHIFT
000202  160067  000000G                         SUB     R0,$SLTMA       ;   DIFFERENTIAL

000206  160200                                  SUB     R2,R0           ; ADD IN CETAB ALLOCATION ADDRESS
000210  016701  000000G                         MOV     $PDVTA,R1       ; ADDR. OF PDV ADDR. TABLE
000214  016702  000000G                         MOV     $PDVNM,R2       ; NUMBER OF ENTRIES
                                        ;
                                        ; RELOCATE THE PDV ADDRESSES, ASSUME PDV TABLE AT FRONT OF IMAGE !
                                        ;
000220  160021                          10$:    SUB     R0,(R1)+        ; SUBTRACT SHIFT BIAS
000222  005302                                  DEC     R2
000224  003375                                  BGT     10$             ;  AND LOOP TILL DONE

000226  014105                                  MOV     -(R1),R5        ; SAVE ADDRESS OF LAST PDV IN CETAB
000230  062705  000000G                         ADD     #Z.LEN,R5       ; ADDRESS OF LAST WORD IN CETAB+2
                                                                        ; ASSUME LAST PDV NOT AN LLC WITH CHANNEL TABLE

                                                .IF DF  R$$MPL                                                  ;[MP01]
                                                ADD     #L.KRB,R5       ; INCLUDE LAST CTB (WITHOUT KRB TABLE)  ;[MP01]
                                                .ENDC                                                           ;[MP01]

000234  162705  000000G                         SUB     #$CETAB,R5      ; CALCULATE SIZE OF CETAB IN BYTES
000240  020504                                  CMP     R5,R4           ; IS INTERNAL BUFFER TOO SMALL ?
000242  003106                                  BGT     35$             ; YES
000244  010567  000000G                         MOV     R5,$CETLN       ; SAVE CETAB LENGTH
000250  020567  000000G                         CMP     R5,$RSIZE       ; DID WE ALREADY READ IN ALL OF CETAB ?
000254  003445                                  BLE     12$             ; IF LE, YES
000256  005067  000000G                         CLR     $BIGCT          ; INIT LARGE CETAB FLAG
                                        ;
                                        ; READ IN REMAINDER OF CETAB
                                        ;
000262  012746  000000G                 40$:    MOV     #$CETAB,-(SP)   ; ORIGINAL ADDRESS
000266  066716  000000G                         ADD     $RSIZE,(SP)     ; PLUS SIZE READ
000272  016701  000000G                         MOV     $BIGCT,R1       ; IS THIS CETAB LARGER THAN 2000 BYTES
```

```
                                        ; REWRITE CEXCM
                                        ;
000122                                          SWTCHO  #$PDVTA         ; POINT AT BUFFER
000130  012702  000000C                         MOV     #<$CEXND-$PDVTA>,R2 ; COMPUTE LENGTH OF TRANSFER
000134                                          PUTRC   .PDVTA,R2       ; WRITE CEXCM BACK
                                        ;
                                        ; REWRITE THE CETAB IMAGE
                                        ;
000152  016700  000000G                         MOV     $CETLN,R0       ; GET LENGTH OF CETAB IMAGE
000156  022700  001000                          CMP     #512.,R0        ; IS IT MORE THAN 1 BLOCK ?
000162  002002                                  BGE     10$             ; IF GE, NO .. OKAY
000164  012700  001000                          MOV     #512.,R0        ; ELSE, LIMIT TRANSFER TO 512. BYTES
000170  160067  000000G                 10$:    SUB     R0,$CETLN       ; CALCULATE LENGTH LEFT TO TRANSFER
000174                                          SWTCHO  #$CETAB         ; POINT OUTPUT BUFFER AT CETAB IMAGE
000202                                          PUTRC   $PDVTA,R0       ; WRITE OUT CETAB IMAGE
000220  005767  000000G                         TST     $CETLN          ; ANY MORE TO TRANSFER ?
000224  003452                                  BLE     20$             ; IF LE, NO
000226                                          SWTCHO  #$CETAB+1000    ; UPDATE OUTPUT BUFFER
000234  066700  000000G                         ADD     $PDVTA,R0       ; AND NEW IMAGE ADDRESS
000240  016701  000000G                         MOV     $CETLN,R1       ; SAVE LENGTH
000244  005067  000000G                         CLR     $BIGCT          ; INIT THIRD WRITE FLAG
000250  022767  001000  000000G                 CMP     #512.,$CETLN    ; IS A THIRD WRITE NEEDED ?
000256  002004                                  BGE     11$             ; IF NOT - BRANCH
000260  012701  001000                          MOV     #512.,R1        ; SET MAXIMUM TRANSFER SIZE
000264  005267  000000G                         INC     $BIGCT          ; SET FLAG FOR THIRD TRANSFER
000270                                  11$:    PUTRC   R0,R1           ; WRITE OUT CETAB IMAGE

000304  005767  000000G                         TST     $BIGCT          ; THIRD TRANSFER NEEDED ?
000310  001420                                  BEQ     20$             ; IF NO - BRANCH
000312                                          SWTCHO  #$CETAB+2000    ; UPDATE OUTPUT BUFFER
000320  062700  001000                          ADD     #512.,R0        ; UPDATE IMAGE ADDRESS
000324  005067  000000G                         CLR     $BIGCT          ; INIT THIRD TRANSFER FLAG
000330  160167  000000G                         SUB     R1,$CETLN       ; SET UP LAST TRANSFER LENGTH
000334                                          PUTRC   R0,$CETLN       ; WRITE OUT LAST PART OF CETAB IMAGE

                                        ;
                                        ; REWRITE NETWORK POOL HOME BLOCK
                                        ;
000352                                  20$:    SWTCHO  #$NTLHB         ; POINT AT BLOCK

000360  005767  000000G                         TST     .MGMGE          ; MEMORY MANAGEMENT SYSTEM ?
000364  100414                                  BMI     30$             ; NO

000366                                          PUTAD   #0,#64.,$NTLPT  ; WRITE IT
000414  000410                                  BR      40$
000416                                  30$:    PUTRC   $NTLPT,#100     ; WRITE IT BACK TO SYSTEM IMAGE
                                        ;
                                        ; REWRITE CEX POOL LISTHEAD IMAGE
                                        ;
000436                                  40$:
                                                .IF NDF R$$MPL                                          ;[MP01]
000436                                          SWTCHO  #$CEAVL
000444                                          PUTRC   .CEAVL,#2       ; REWRITE CEX POOL LISTHEAD
000464                                          SWTCHO                  ; SWITCH TO DEFAULT BUFFER
                                                .ENDC                                                   ;[MP01]

000472  042767  000000G 000000G                 BIC     #FE.CEX,$FMASK  ; INDICATE COMEXEC NOT CORE RESIDENT
```

```
                                            BIS     #DF.SCO,$DFLAG  ; SET LOWCR FLAG                        ;[MP01]
                                            GETRV   #$SVCOL,#2      ; READ VBN OF DISPLACED MEMORY          ;[MP01]
                                            SWTCHI  #$VCTOF         ; SET UP INPUT ADDRESS                  ;[MP01]
                                            BIS     #DF.SCO,$DFLAG  ; SET LOWCR FLAG                        ;[MP01]
                                            GETRV   #$SVCOL+2,#2    ; READ OFFSET INTO FIRST DISPLACED BLK  ;[MP01]
                                            SWTCHI                                                          ;[MP01]
                                            .ENDC                                                           ;[MP01]

 000602                             100$:   RETURN


                                    ;
                                    ; ERROR RETURNS
                                    ;

 000604  012746  000202'            110$:   MOV     #VFILE,-(SP)    ; SET RETURN ADDRESS
 000610                                     ERRPT$  ERR1,$EROUT     ; ISSUE WARNING MESSAGE
 000620  012746  000202'            122$:   MOV     #VFILE,-(SP)    ; SET RETURN ADDRESS
 000624                                     ERRPT$  ERR2,$EROUT     ; ISSUE WARNING MESSAGE
 000634  052767  000000G 000000G    130$:   BIS     #V1.NSY,$VFLG1  ; INDICATE NOT A SYSTEM IMAGE FILE
 000642                                     ERRPT$  ERR3,$EROUT,FATAL ; FILE NOT A SYSTEM IMAGE FILE
 000656                             132$:   ERRPT$  ERR13,$EROUT,FATAL ; CORRUPT SYSTEM IMAGE

                                    ;
                                    ; ERROR TEXT
                                    ;

 000672                                     ERMSG$  <M-PLUS system not saved%n>
 000723                                     ERBLK$  ERR0
 000726                                     ERMSG$  <Error parsing file name - ( %A )>
 000766                                     ERBLK$  ERR1,<FILE>
 000774                                     ERMSG$  <Open failure on system image file - ( %E )%n>
 001050                                     ERMSG$  <       ( %A )>
 001057                                     ERBLK$  ERR2,<SYSFDB+F.ERR,FILE>
 001066                                     ERMSG$  <Supposed system image file not actually a system image.>
 001155                                     ERMSG$  <%n     ( %A )>
 001166                                     ERBLK$  ERR3,<FILE>
 001174                                     ERMSG$  <System image file corrupted because VNP was terminated%n>
 001264                                     ERMSG$  <abnormally.             ( %A )>
 001307                                     ERBLK$  ERR13,<FILE>

                                    ;
                                    ; LOCAL DATA
                                    ;

                                            .NLIST BEX
 001314     012     015     105     SYSPMT: .ASCIZ  <12><15>'Enter filename: '
                                            .EVEN
                                            .LIST BEX

```

```
004010  004022' 000000          PPRO:   .WORD   NETPF,0         ; PCB PROTECTION WORD OFFSET
004014  064160  071330                  .RAD50  "P.PRO"
004020  000000                          .WORD   0
004022  004034' 000000          NETPF:  .WORD   NTUIC,0         ; NETWORK POWERFAIL RECOVERY
004026  125365  077606                  .RAD50  "$NETPF"
004032  000000                          .WORD   0
004034  000000  000000          NTUIC:  .WORD   0,0             ; NETWORK UIC
004040  125404  102253                  .RAD50  "$NTUIC"
004044  000000                          .WORD   0

                                ; COMEXEC SYMBOLS

                                        .IF NDF R$$MPL                                          ;[MP01]
004046                          CEXSTB:                         ; REFERENCE POINT

004046                          PDVTA:
004046  004060' 000000                  .WORD   CEAVL,0         ; STARTING ADDRESS OF PDVS
004052  125504  106241                  .RAD50  "$PDVTA"
004056  000000                          .WORD   0
004060  004072' 000000          CEAVL:  .WORD   DLCOV,0         ; CEX POOL LISTHEAD
004064  124475  004674                  .RAD50  "$CEAVL"
004070  000000                          .WORD   0

                                        .IFF                                                    ;[MP01]
                                CEXSTB:                                                         ;[MP01]
                                        .ENDC                                                   ;[MP01]
004072  004104' 000000          DLCOV:  .WORD   CRCOV,0         ; DLC PROTOCOL OVERHEAD
004076  015343  125456                  .RAD50  "DLC$OV"
004102  000000                          .WORD   0
004104  004116' 000000          CRCOV:  .WORD   CEPWR,0         ; CRC OVERHEAD
004110  012623  125456                  .RAD50  "CRC$OV"
004114  000000                          .WORD   0
004116  000000  000000          CEPWR:  .WORD   0,0             ; COMM/EXEC POWERFAIL RECOVERY ROUTINE
004122  124475  063652                  .RAD50  "$CEPWR"
004126  000000                          .WORD   0

```

```
		.TITLE	VLBIO - VNP LIBRARY BLOCK I/O ROUTINES
		.IDENT	/V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;	VNP - LIBRARY BLOCK I/O ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00	27-FEB-78
;	VERSION 2.0 RELEASE
;
; 2.00	14-DEC-79
;	DECNET-11M/S V3.0
;	DECNET-11M-PLUS V1.0
;
; 3.00	16-APR-82
;	DECNET-11M V3.1
;	DECNET-11M-PLUS V1.1
;
; 4.00	07-NOV-83
;	DECNET-11M V4.0
;	DECNET-11M-PLUS V2.0
;
; 5.00	22-JUL-85
;	DECnet-11M/S V4.2
;	DECnet-11M-Plus V3.0
;	DECnet-Micro/RSX V1.0
;
;
```

```
                                        ;
                                        ; Get command line/set function code/fill in data provided with command
                                        ;

000000  000240                          $VNCP:: NOP
000002  005067  000014'                 1$:     CLR     EBUF            ; Indicate no error yet
000006  005067  000000G                         CLR     $OPTON          ; Initialize parse variables
000012  005067  000000G                         CLR     $CMAND          ;       ...
000016  005067  000000G                         CLR     $QUAL           ;       ...
000022  005067  000000G                         CLR     $TYPE           ;       ...
000026  012700  000000G                         MOV     #RQB,R0         ; Zero out the entire request block
000032  012701  000050                          MOV     #40.,R1         ;       ...
000036  005020                          10$:    CLR     (R0)+           ;       ...
000040  005301                                  DEC     R1              ;       ...
000042  001375                                  BNE     10$             ;       ...
000044  042767  000000C 000000G                 BIC     #^C<VF.SS!VF.INI>,$VFLAG ; Clear transitory states
000052  012701  001400                          MOV     #<3*400>,R1     ; Match at least 3 chars
000056  012702  000000G                         MOV     #KEYTAB,R2      ; Point at keyword table
000062  012705  000000G                         MOV     #PARTAB,R5      ; Point at state table
000066                                          CALL    $GTPLN          ; Get a parsed line
000072  103107                                  BCC     20$             ; If cc, no problem
000074  102467                                  BVS     15$             ; If vs, gcl error
                                        ;
                                        ; Syntax error
                                        ;
000076                                          SAVRG   <R3,R4>         ; Save parse pointers.
000102  012767  000106' 000044'                 MOV     #CRLF,GBUF      ; Start a new line.
000110  012767  000006  000046'                 MOV     #LCRLF,GLEN
000116                                          DIR$    #GMSG

000124  016767  000000G 000044'                 MOV     $CLBUF,GBUF     ; Point to beg of cmd line.
000132  011667  000046'                         MOV     (SP),GLEN
000136  166767  000000G 000046'                 SUB     $CLBUF,GLEN     ; Length of parsed portion.
000144  003403                                  BLE     12$             ; If anything was parsed,
000146                                            DIR$  #GMSG           ; Print parsed portion of cmd line.
000154                                  12$:
000154  012767  000104' 000044'                 MOV     #MARKER,GBUF    ; Point to marker string.
000162  012767  000002  000046'                 MOV     #LMARKER,GLEN
000170                                          DIR$    #GMSG           ; Mark unparsed portion.

000176                                          RESRG   <R4,R3>
000202  010400                                  MOV     R4,R0           ; Find end of line,
000204  060300                                  ADD     R3,R0
000206  112720  000015                          MOVB    #<15>,(R0)+     ;  and insert cr.
000212  005203                                  INC     R3              ; Update char count.
000214  010467  000044'                         MOV     R4,GBUF
000220  010367  000046'                         MOV     R3,GLEN
000224                                          DIR$    #GMSG           ; Print unparsed portion of cmd line.

000232                                          ERROR$  SYNERR,WARN     ; Say "Syntax error".
000252  000653                                  BR      1$              ; Get next line.
                                        ;
                                        ; Check GCL error
                                        ;
000254  012767  000011  000000G         15$:    MOV     #LS.XIT,RQBST   ; Assume exit function
000262  122777  000000G 000000G                 CMPB    #IE.EOF,@$CLIOS ; Just an end-of-file ?
000270  001413                                  BEQ     EOJ             ; If eq, yes .. okay
```

```
003566                          101$:    ERROR$   ILLPRI


                                ;
                                ; Alias scope device name
                                ;
003606                          $SCOP1::
003606  116767  000000G 000060'          MOVB     .PCHAR,SCODV      ; Store device name
003614                                   RETURN
003616                          $SCOP2::
003616  116767  000000G 000061'          MOVB     .PCHAR,SCODV+1    ; Second half of device name
003624                                   RETURN
003626                          $SCOP3::
003626  016767  000000G 000062'          MOV      .PNUMB,SCOUN      ; Store unit number
003634                                   RETURN
003636                          $SCOND::
003636  016767  000000G 000000G          MOV      $DEVHD,$BFR       ; Get start of device list
003644  016700  000060'         10$:     MOV      SCODV,R0          ; Get scope device name
003650                                   CALL     FNDDCB            ; Try to find the DCB
003654  103425                           BCS      101$              ; If cs, no luck
003656  016700  000062'                  MOV      SCOUN,R0          ; Pick up scope unit number
003662  120067  000000C                  CMPB     R0,$BFR+D.UNIT    ; Is this the right DCB ?
003666  103766                           BLO      10$               ; If lo, no .. keep looking
003670  120067  000000C                  CMPB     R0,$BFR+D.UNIT+1  ; Really ?
003674  101363                           BHI      10$               ; If hi, no .. keep looking
003676  005046                           CLR      -(SP)             ; Get startin unit number
003700  116716  000000C                  MOVB     $BFR+D.UNIT,(SP)  ; ...
003704  162600                           SUB      (SP)+,R0          ; Minus starting unit number
003706  016701  000000C                  MOV      $BFR+D.UCBL,R1    ; Get length of each UCB
003712                                   CALL     $MUL              ; Get offset to prooer UCB
003716  066701  000000C                  ADD      $BFR+D.UCB,R1     ; Point to UCB
003722  010167  000016G                  MOV      R1,RQB+LR.UCB     ; Save UCB address
003726                                   RETURN
003730                          101$:    ERROR$   NSTERM            ; Terminal not in system

                                ;
                                ; Secondary csr value
                                ;
003750  005767  000000G         $SCSR::  TST      .PNUMH            ; Must be single precision
003754  001014                           BNE      101$              ;       ...
003756  022767  160000  000000G          CMP      #160000,.PNUMB    ; An i/o page address
003764  101010                           BHI      101$              ;       ...
003766  032767  000003  000000G          BIT      #3,.PNUMB         ; And a multiple of 4
003774  001004                           BNE      101$              ;       ...
003776  016767  000000G 000016G          MOV      .PNUMB,RQB+LO.SCR ; Set secondary csr value
004004                                   RETURN
004006                          101$:    ERROR$   ILLUCS

                                ;
                                ; NSP segment size>
                                ;
004026  016700  000000G         $SEG::   MOV      $RDBSZ,R0         ; GET RDB SIZE
004032  162700  000022                   SUB      #18.,R0           ; MINUS NSP OVERHEAD
004036  005767  000000G                  TST      .PNUMB            ; CHECK LOWER LIMIT
004042  003407                           BLE      101$              ; IF ERROR - BRANCH
004044  026700  000000G                  CMP      .PNUMB,R0         ; CHECK HIGH LIMIT
004050  002004                           BGE      101$              ; IF ERROR - BRANCH

```

```
B  1
C  1
D  1
E  1
F  1
G  1
H  1
I  1
J  1
K  1
L  1
M  1
N  1
B  2
C  2
D  2
E  2
F  2
G  2
H  2
I  2   VFPRDT
J  2
K  2
L  2
M  2
N  2
B  3
C  3
D  3
E  3
F  3
G  3
H  3
I  3
J  3
K  3
L  3   VFPRNW
M  3
N  3
B  4
C  4
D  4
E  4
F  4
G  4
H  4
I  4
J  4
K  4
L  4
M  4
N  4
B  5
C  5
D  5
E  5
F  5
G  5
H  5
I  5
```

```
J  5
K  5
L  5
M  5
N  5
B  6   VFPX29
C  6
D  6
E  6
F  6
G  6
H  6
I  6
J  6
K  6
L  6
M  6
N  6
B  7   VFPX3P
C  7
D  7
E  7
F  7
G  7
H  7
I  7
J  7
K  7
L  7
M  7
N  7
B  8
C  8
D  8
E  8
F  8   VHDAT
G  8
H  8
I  8
J  8
K  8
L  8   VHELP
M  8
N  8
B  9
C  9
D  9
E  9
F  9
G  9
H  9
I  9
J  9
K  9   VIMI
L  9
M  9
N  9
B 10
C 10
D 10
```

```
E 10
F 10
G 10
H 10
I 10
J 10
K 10   VIMO
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11   VINI
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13   VLBIO
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14   VNCP
I 14
J 14
K 14
L 14
```

```
M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16
```

```
                                              .SBTTL  STORID - Store id-string according to system type              ;[TD001]
                                    ;+                                                                               ;[TD001]
                                    ; *** STORID - Store id-string according to system type                         ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ; FUNCTION                                                                       ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ;         This routine is used to check and store various parameters            ;[TD001]
                                    ;         during the parsing of VNP commands. In PSI V2.0 compatible            ;[TD001]
                                    ;         systems, it checks for and stores parameters in PSI V2.0              ;[TD001]
                                    ;         compatible format. In extended systems, id-strings of up              ;[TD001]
                                    ;         to MAXID characters are permitted and stored appropriately.           ;[TD001]
                                    ;         The buffer is padded with spaces. The length returned in the          ;[TD001]
                                    ;         buffer is the maximum length for old-format systems (fixed            ;[TD001]
                                    ;         fields) or the actual length for new-format systems (variable         ;[TD001]
                                    ;         fields).                                                               ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ; INPUTS                                                                         ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ;         R0      - address of buffer to contain string prefixed by length      ;[TD001]
                                    ;         R1      - maximum length of string in old-format system               ;[TD001]
                                    ;                   (R1 < MAXID)                                                 ;[TD001]
                                    ;         .PSTCN  - actual length of string                                      ;[TD001]
                                    ;         $PFLAG  - PSI system type flags                                        ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ; OUTPUTS                                                                        ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ;         R0, R1, R2 destroyed                                                   ;[TD001]
                                    ;         String and length stored in buffer                                     ;[TD001]
                                    ;         Carry set if invalid string length for system type                     ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ;-                                                                               ;[TD001]
006416                              STORID:                                                                          ;[TD001]
006416  010046                                MOV     R0,-(SP)                ; Save buffer address                  ;[TD001]
006420  010146                                MOV     R1,-(SP)                ; Save maximum length                  ;[TD001]
006422  105020                                CLRB    (R0)+                   ; Clear length field in output         ;[TD001]
006424  012701  000020                        MOV     #MAXID,R1               ; Always clear MAXID spaces            ;[TD001]
006430                                        CALL    SPFILL                                                         ;[TD001]
006434  012601                                MOV     (SP)+,R1                ; Restore length                       ;[TD001]
006436  012602                                MOV     (SP)+,R2                ; Get back buffer address              ;[TD001]
006440  032767  000000G 000000G               BIT     #PF.XDF,$PFLAG          ; Check for extened data format        ;[TD001]
006446  001005                                BNE     20$                                                            ;[TD001]
                                    ;                                                                                ;[TD001]
006450  026701  000000G                       CMP     .PSTCN,R1               ; Old format, so use max length in R1  ;[TD001]
006454  101020                                BHI     99$                     ; HI, then string too long             ;[TD001]
006456  110122                                MOVB    R1,(R2)+                ; Set length to maximum for fixed field ;[TD001]
006460  000406                                BR      30$                                                            ;[TD001]
                                    ;                                                                                ;[TD001]
006462  026727  000000G 000020      20$:      CMP     .PSTCN,#MAXID           ; New format, max length is MAXID      ;[TD001]
006470  101012                                BHI     99$                                                            ;[TD001]
006472  116722  000000G                       MOVB    .PSTCN,(R2)+            ; Set actual length for variable field ;[TD001]
                                    ;                                                                                ;[TD001]
                                    ; Move string to buffer                                                          ;[TD001]
                                    ;                                                                                ;[TD001]
006476  016701  000000G             30$:      MOV     .PSTCN,R1                                                      ;[TD001]
006502  016700  000000G                       MOV     .PSTPT,R0               ; Get address of source                ;[TD001]
006506                                        CALL    MOVE                    ; Move string to buffer                ;[TD001]
006512  000241                                CLC                             ; Signal success                       ;[TD001]
```

```
                                                                   B  2
VNCP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 6
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE               REFERENCES

OF$INS  = 000001            #5-61
OF$LOG  = 000100            #5-61
OF$OFF  = 000002            #5-61
OF$ON   = 000000            #5-61
OF$SMC  = 000200            #5-61
ONAMTL    001151 R          #5-164        5-164       7-842
OP$ADD  = 000200            #5-61
OP$ALL  = 000200            #5-61
OP$CCS  = 000004            #5-61
OP$CIE  = 004000            #5-61         7-385
OP$CON  = 000400            #5-61
OP$COP  = 000001            #5-61
OP$COS  = 000100            #5-61
OP$DST  = 000200            #5-61
OP$DTE  = 000040            #5-61
OP$DUP  = 000002            #5-61
OP$EST  = 000100            #5-61
OP$EVE  = 010000            #5-61
OP$FIL  = 000200            #5-61
OP$GRP  = 000100            #5-61
OP$HOS  = 000001            #5-61
OP$HTM  = 000002            #5-61
OP$INA  = 004000            #5-61
OP$INC  = 001000            #5-61
OP$KDS  = 002000            #5-61
OP$KDT  = 000400            #5-61
OP$KEV  = 001000            #5-61
OP$KGR  = 001000            #5-61
OP$KNT  = 010000            #5-61
OP$KSK  = 002000            #5-61
OP$LCT  = 000200            #5-61
OP$LNE  = 000040            #5-61          7-385
OP$LOC  = 000001            #5-61         #5-61
OP$LST  = 000002            #5-61
OP$LTM  = 000001            #5-61
OP$LVL  = 000010            #5-61
OP$MAC  = 000010            #5-61
OP$MCO  = 000002            #5-61
OP$MDE  = 000400            #5-61
OP$MLI  = 000004            #5-61
OP$MON  = 000100            #5-61
OP$MST  = 000020            #5-61
OP$NAM  = 000004            #5-61         #5-61       #5-61
OP$NET  = 004000            #5-61
OP$NLI  = 000002            #5-61
OP$NOD  = 000020            #5-61          7-385
OP$OUT  = 002000            #5-61
OP$OWN  = 000020            #5-61
OP$PAR  = 000010            #5-61
OP$PRI  = 000010            #5-61
OP$RET  = 010000            #5-61
OP$RPA  = 000010            #5-61
```

```
                                                                   C  2
VNCP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 7
```

```
                                              .SBTTL  $PANIC - VNP EXIT AND PANIC EXIT ROUTINE
                                      ;+
                                      ; **-$PANIC-ALL EXITS GO THROUGH HERE
                                      ;
                                      ;-

 000130   000167   000000G            $PANIC::JMP     $VXIT           ; EXIT AS GRACEFULLY AS POSSIBLE
```

```
    613 002014  000241                        CLC                           ; INDICATE SUCCESS
    614 002016                        30$:    RETURN
    615
    616                               ;
    617                               ; ERROR CONDITION
    618                               ;
    619 002020                        101$:
    620 002020                                CALLR   $XPRO               ; DO A PARTIAL "CLEAR PROCESS"
```

```
                                                                   B  5
VNP         CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 5

SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE               REFERENCES

QF$LOO  = 000010            #5-69
RQB     = ******   GX        9-174     10-222     10-229     10-236    *10-242    *10-243    *10-244    *10-245     10-246
                            *10-248   *11-298     12-340     12-347     12-354    *12-360    *12-361    *12-362    *12-363
                            *12-370   *12-371       -372    *12-374     15-514     15-518     15-523     15-530     15-542
                             15-544    15-544     , -544
R$$EIS  = ******             15-526
R$$MPL  = ******             5-62      5-71       19-756
R$$11D  = ******             15-526
SC        001572 R           15-516    15-520     15-528     15-532     15-542     15-544    #15-548
SET       001600 R           15-516    15-528    #15-530
SYSFDB  = ******   GX       *20-790
TY$CHA  = 000400            #5-69
TY$EVE  = 001000            #5-69
TY$STA  = 002000            #5-69
TY$SUM  = 000000            #5-69
UNS1      000650 R           11-306   #11-310
VF.ECH  = ******   GX        11-288
VF.NTI  = ******   GX        11-299
WC.CNT  = 000400            #5-69      10-208     10-227     12-328     12-345
WC.EVF  = 004000            #5-69
WC.TRI  = 002000            #5-69      12-328
WC.UNI  = 001000            #5-69      10-208     10-234     12-328     12-352
ZF.MFL  = ******   GX        16-612
ZF.PSE  = ******   GX        16-577
Z.FLG   = ******   GX        16-577   *16-612     20-780
Z.NAM   = ******   GX        10-222    10-242     12-340     12-360     16-589     16-605     20-784
Z.PCB   = ******   GX        16-571    20-785
$ALADJ  = ******   GX        16-598
$ALECL  = ******   GX        16-607
$ALMHC  = ******   GX        16-601
$ALTLC  = ******   GX        19-767
$BFR    = ******   GX        20-791
$BIAS   = ******   GX       *20-787   *20-791
$CACH   = ******   GX        16-594
$CCEX   = ******   GX        13-438
$CKT    = ******   GX        16-596
$CLFLG  = ******   GX       *11-279
$CLIN   = ******   GX        13-440
$CLSAV  = ******   GX        11-278
$CNOD   = ******   GX        13-447
$CPRO   = ******   GX        13-439
$CSRLD  = ******   GX        19-763
$DALI   = ******   GX        13-458
$DCIR   = ******   GX        13-463
$DDMPV  = ******   GX        17-650    19-746
$DLCPV  = ******   GX        19-734
$DLIN   = ******   GX        13-455
$DLOG   = ******   GX        13-459
$DMOD   = ******   GX        13-465
$DNOD   = ******   GX        13-457
$DOBJ   = ******   GX        13-460
$DPRO   = ******   GX        13-456
```

```
                                                                   C  5
VNP         CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 6
```

Table of contents

```
    70- 3224     ACTION ROUTINE: MAXIMUM NUMBER OF NODES SEEN
    71- 3247     ACTION ROUTINE: MAXIMUM COST SEEN
    72- 3270     ACTION ROUTINE: MAXIMUM NUMBER OF HOPS
    73- 3295     ACTION ROUTINE: ROUTING TIMER
    74- 3313     ACTION ROUTINE: MAXIMUM AREA
    75- 3338     ACTION ROUTINE: BROADCAST ENDNODES
    76- 3384     ACTION ROUTINE: AREA MAX COST
    77- 3428     ACTION ROUTINE: EVENT CLASS NUMBER FOUND
    78- 3453     ACTION ROUTINE: EVENT TYPE MASK SEEN
    79- 3473     ACTION ROUTINE: SET UP FILTER CONTROL BLOCK
    80- 3519     FINFIL - FIND PLACE TO INSERT FILTER BLOCK
    81- 3549     ACTION ROUTINE: SET LOGGING STATE
    82- 3568     ACTION ROUTINE: OBJECT NUMBER SEEN
    83- 3591     ACTION ROUTINE: OBJECT NAME SEEN
    84- 3617     ACTION ROUTINE: VERIFICATION LEVEL SEEN
    85- 3635     ACTION ROUTINE: OBJECT FLAGS SEEN
    86- 3655     ACTION ROUTINE: OBJECT COPIES SEEN
    87- 3705     FINOBJ - FIND PLACE TO INSERT OBJECT
    88- 3732     FINREM - FIND PLACE FOR A REMOTE NODE BLOCK
    89- 3771     QUICK - FAST REMOTE NODE PROCESSING
    90- 3805     FIL$DF/SER$DF ACTION ROUTINES
   102- 4417     CRFNBK - CREATE FILENAME DESCRIPTOR BLOCK
   103- 4476     .FLBKO - FILE NAME BLOCK PROCESSING
   105- 4685     QINS - INSERT ENTRY INTO QUEUE
   106- 4717     PERFORM 'MARKED FOR LOAD' SCAN
   107- 4764     REQUEST NETLOADER TO LOAD A LINE
   108- 4796     REQUEST NETLOADER TO LOAD A DLM CIRCUIT
   109- 4839     REQUEST NETLOADER TO LOAD A PROCESS
   110- 4867     REQUEST NETLOADER TO RUN
```

```
                                              .SBTTL  NTWARN - NTINIT WARNING HANDLING
                                      ;+
                                      ; *** - NTWARN - WARNING HANDLING
                                      ;
                                      ; INPUTS:
                                      ;       R0 = NUMBER TO BE PRINTED
                                      ;       2(SP) = ERROR NUMBER
                                      ;
                                      ; OUTPUTS:
                                      ;       MESSAGE PRINTED ON TERMINAL
                                      ;       ALL REGISTERS PRESERVED
                                      ;
                                      ;-

001554                                NTWARN: .
001554  010067  000220                        MOV     R0,SBUF+2               ; SAVE NUMBER
001560  016667  000002  000210                MOV     2(SP),SBUF              ; GET ERROR NUMBER
001566                                        CALL    $SAVAL                  ; SAVE ALL REGISTERS
001572  012767  001752' 000222                MOV     #PREWRN,QIO+Q.IOPL      ; ADDRESS OF PREFIX
001600  012767  000024  000216                MOV     #PREWLN,QIO+Q.IOPL+2    ; LENGTH OF PREFIX MESSAGE
001606  012767  000044  000212                MOV     #'$,QIO+Q.IOPL+4        ; CARRIAGE CONTROL
001614                                        DIR$    #QIO                    ; PRINT PREFIX MESSAGE
001622  116700  000150                        MOVB    SBUF,R0                 ; GET ERROR NUMBER
001626  006300                                ASL     R0                      ; CONVERT TO DOUBLE WORD INDEX
001630  006300                                ASL     R0                      ; ...
001632  062700  002032'                       ADD     #ERRTAB-4,R0            ; POINT INTO TABLE
001636  012067  000160                        MOV     (R0)+,QIO+Q.IOPL        ; STORE ADDRESS OF MESSAGE
001642  011067  000156                        MOV     (R0),QIO+Q.IOPL+2       ; AND LENGTH
001646  005067  000154                        CLR     QIO+Q.IOPL+4            ; NO CARRIAGE CONTROL
001652                                        DIR$    #QIO                    ; PRINT MESSAGE
001660  012700  001776'                       MOV     #SBUF,R0                ; POINT AT SCRATCH BUFFER
001664  010003                                MOV     R0,R3                   ; SAVE START OF BUFFER
001666  016001  000002                        MOV     2(R0),R1                ; GET NUMBER TO CONVERT
001672  005002                                CLR     R2                      ; ZERO SUPPRESSION
001674                                        CALL    $CBDSG                  ; CONVERT TO SIGNED DECIMAL
001700  112720  000056                        MOVB    #<'.>,(R0)+             ; INSERT PERIOD
001704  160300                                SUB     R3,R0                   ; CALCULATE LENGTH OF BUFFER
001706  010367  000110                        MOV     R3,QIO+Q.IOPL           ; SET ADDRESS OF BUFFER
001712  010067  000106                        MOV     R0,QIO+Q.IOPL+2         ; AND LENGTH
001716  012767  000053  000102                MOV     #'+,QIO+Q.IOPL+4        ; CARRIAGE RETURN
001724                                        DIR$    #QIO                    ; PRINT NUMBER
001732                                        RETURN


                                      ;
                                      ; PREFIX MESSAGE
                                      ;
001734     126     116     120        PREFIX: .ASCII  "VNP -- *DIAG* "
001737     040     055     055
001742     040     052     104
001745     111     101     107
001750     052     040
        000016                                PRELEN=.-PREFIX
001752     126     116     120        PREWRN: .ASCII  "VNP -- *WARNING* -- "
001755     040     055     055
001760     040     052     127
001763     101     122     116
001766     111     116     107
```

```
   1137                                            .SBTTL  STATE TRANSITIONS FOR ESTIMATING NTPOOL SIZE
   1138
   1139 003632                                     ISTAT$  IBUF,KBUF
   1140
   1141 003632                                     STATE$
   1142 003632                                      TRAN$   %BUF$DF%,,,1,CFSFLG    ; SIGNAL THAT MACRO WAS FOUND
   1143 003632                                     STATE$
   1144 003632                                      TRAN$   $NUMBR                 ; NUMBER OF CCBS
   1145 003632                                     STATE$
   1146 003632                                      TRAN$   <',>
   1147 003632                                     STATE$
   1148 003632                                      TRAN$   $NUMBR                 ; SIZE CCBS
   1149 003632                                     STATE$
   1150 003632                                      TRAN$   <',>
   1151 003632                                     STATE$
   1152 003632                                      TRAN$   $NUMBR,,.N1            ; NUMBER OF RDBS
   1153 003632                                     STATE$
   1154 003632                                      TRAN$   <',>
   1155 003632                                     STATE$
   1156 003632                                      TRAN$   $NUMBR,,.N2            ; SIZE OF RDBS
   1157 003632                                     STATE$
   1158 003632                                      TRAN$   <',>
   1159 003632                                     STATE$
   1160 003632                                      TRAN$   $NUMBR,,.N1            ; NUMBER OF SDBS
   1161 003632                                     STATE$
   1162 003632                                      TRAN$   <',>
   1163 003632                                     STATE$
   1164 003632                                      TRAN$   $NUMBR,,.N3            ; SIZE OF SDBS
   1165 003632                                     STATE$
   1166 003632                                      TRAN$  <',>
   1167 003632                                     STATE$
   1168 003632                                      TRAN$   $NUMBR                 ; RDB THRESHOLD
   1169 003632                                     STATE$
   1170 003632                                      TRAN$  <',>
   1171 003632                                     STATE$
   1172 003632                                      TRAN$   $NUMBR,$EXIT,.N4       ; MAX DLC OVERHEAD
   1173 003632                                     STATE$
   1174
```

```
                                          .SBTTL  CFGMFL - CETAB SCANNER FOR NETWORK INITIALISER
                                          .SBTTL  DEFINE LOCAL DATA AREAS

                                  ;+
                                  ;       DEFINE TYPE CODES FOR MACROS
                                  ;-

          000001                  MC.SLT=1                      ; SLT$DF  SEEN
          000002                  MC.LLC=2                      ; LLC$DF
          000003                  MC.DLC=3                      ; DLC$DF
          000004                  MC.DDM=4                      ; DDM$DF
          000010                  MC.NOD=10                     ; NOD$DF
          000020                  MC.OBJ=20                     ; OBJ$DF
          000040                  MC.REM=40                     ; REM$DF
          000100                  MC.ROU=100                    ; ROU$DF
          000200                  MC.ADD=200                    ; ADD$DF
          000400                  MC.LOG=400                    ; LOG$ST
          001000                  MC.EVT=1000                   ; EVT$DF
          002000                  MC.STA=2000                   ; STA$DF
                                  ;

                                  ;+
                                  ;       DEFINE SPECIAL CHARACTERS WE CANNOT USE IN TPARS DEFINITIONS
                                  ;-

          000074                  LBRA='<
          000076                  RBRA='>
                                  ;
```

```
                                .SBTTL  NOD$DF PARSE TABLES
                        ;+
                        ;       STATE TRANSITIONS FOR THE 'NOD$DF' MACRO
                        ;-

006324                          STATE$  NODIF
006324                           TRAN$   LBRA,,.NODNM
006324                          STATE$
006324                           TRAN$   !STRING
006324                          STATE$
006324                           TRAN$   <',>
006324                          STATE$
006324                           TRAN$   LBRA,,.SYSID
006324                          STATE$
006324                           TRAN$   !STRING
006324                          STATE$
006324                           TRAN$   <',>
006324                          STATE$
006324                           TRAN$  $NUMBR,,.NODAR
006324                          STATE$
006324                           TRAN$  <',>
006324                          STATE$
006324                           TRAN$   $NUMBR,,.NODNB
006324                          STATE$
006324                           TRAN$   <',>
006324                          STATE$
006324                           TRAN$  $NUMBR,,.HSTAR
006324                          STATE$
006324                           TRAN$  <',>
006324                          STATE$
006324                           TRAN$   $NUMBR,$EXIT,.HSTAD
```

```
  2461                                          ;+
  2462                                          ; SER$DF STATE TABLE CONTINUED
  2463                                          ;-
  2464
  2465
  2466 006324                                           STATE$  E
  2467 006324                                            TRAN$  <',>,F
  2468 006324                                            TRAN$  $STRNG,,PAS,SF.PSS,SERFLG
  2469
  2470 006324                                           STATE$
  2471 006324                                            TRAN$  $EOS,$EXIT,SEREND
  2472 006324                                            TRAN$  <',>
  2473
  2474 006324                                           STATE$  F
  2475 006324                                            TRAN$  <',>,G
  2476 006324                                            TRAN$  $NUMBR,,DADDR,SF.DPA,SERFL
  2477
  2478 006324                                           STATE$
  2479 006324                                            TRAN$  $EOS,$EXIT,SEREND
  2480 006324                                            TRAN$  <',>
  2481
  2482 006324                                           STATE$  G
  2483 006324                                            TRAN$  <',>,H
  2484 006324                                            TRAN$  $NUMBR,,DCT,SF.DPC,SERFLG
  2485
  2486 006324                                           STATE$
  2487 006324                                            TRAN$  $EOS,$EXIT,SEREND
  2488 006324                                            TRAN$  <',>
  2489
  2490 006324                                           STATE$  H
  2491 006324                                            TRAN$  <',>,I
  2492 006324                                            TRAN$  $STRNG,,HRDADD,SF.HAD,SERFLG
  2493
  2494
  2495 006324                                           STATE$
  2496 006324                                            TRAN$  $EOS,$EXIT,SEREND
  2497 006324                                            TRAN$  <',>
  2498
  2499
  2500 006324                                           STATE$  I
  2501 006324                                            TRAN$  %SF.PH3%,$EXIT,SEREND,SF.PH3,SERFLG
  2502
  2503 006324                                           STATE$
  2504 006324                                            TRAN$  !END
  2505
  2506
  2507
  2508 006324                                           STATE$  SERST
  2509 006324                                            TRAN$  <'.>,MULP
  2510 006324                                            TRAN$  <'->,MUXP
  2511
  2512 006324                                           STATE$  MUXP
  2513 006324                                            TRAN$  $NUMBR,ZAS,SERMUX
  2514
  2515 006324                                           STATE$  MULP
  2516 006324                                            TRAN$  $NUMBR,,SERMUL
  2517
```

```
2744                                          .SBTTL  ACTION ROUTINE: NODE NAME SEEN
2745                                  ;+
2746                                  ;        **-.NODNM-NODE NAME SEEN
2747                                  ;
2748                                  ;        THE NODE NAME HAS BEEN FOUND IN THE 'NOD$DF' MACRO.
2749                                  ;-
2750                                  ;
2751 007056                           .NODNM:
2752 007056                                   SWTCHI  #$HMBUF         ; USE TEMP BUFFER
2753 007064                                   GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK
2754 007104                                   SWTCHI                  ; RESET DEFAULT BUFFER
2755
2756 007112  016767  000000G 001116           MOV     .PSTPT,STRPTR   ; SAVE ADDRESS OF NEXT BYTE
2757 007120  005267  001112                   INC     STRPTR          ;    WE ARE POINTED AFTER THE "<" NOW
2758 007124  012767  000006G 001110           MOV     #$HMBUF+D$LNAM,STRADR ; PLACE TO PUT THE TEXT
2759 007132  012767  000006  001100           MOV     #6,STRMAX       ; MAXIMUM TEXT LENGTH
2760
2761 007140  016746  001074                   MOV     STRMAX,-(SP)    ; GET NAME SIZE
2762 007144  016700  001066                   MOV     STRPTR,R0       ; GET ADDRESS OF NAME
2763 007150  012701  000000G                  MOV     #$BFR,R1        ; GET BUFFER ADDRESS
2764 007154  112021                   5$:     MOVB    (R0)+,(R1)+     ; COPY A CHARACTER
2765 007156  121027  000076                   CMPB    (R0),#'>        ; RIGHT BRACKET ?
2766 007162  001006                           BNE     10$             ; IF NOT - BRANCH
2767 007164  005216                           INC     (SP)            ; do not count >
2768 007166  005316                   7$:     DEC     (SP)            ; COUNT ONE CHARACTER
2769 007170  001405                           BEQ     15$             ; IF DONE - BRANCH
2770 007172  112721  000040                   MOVB    #40,(R1)+       ; SET SPACE
2771 007176  000773                           BR      7$
2772
2773 007200  005316                   10$:    DEC     (SP)            ; COUNT ONE CHARACTER
2774 007202  001364                           BNE     5$              ; IF MORE - BRANCH
2775
2776 007204  005726                   15$:    TST     (SP)+           ; CLEAN STACK
2777 007206                                   SWTCHO
2778 007214                                   PUTRC   .SYSNM,STRMAX   ; UPDATE IMAGE
2779
2780 007234                                   RETURN                  ; WAIT FOR THE ">" (.STRNG)
```

```
                                                .SBTTL  ACTION ROUTINE: MAXIMUM COST SEEN

                                        ;+
                                        ; *** - .RCOST - MAXIMUM COST SEEN
                                        ;
                                        ; INPUT:
                                        ;       .PNUMB = MAXIMUM COST ALLOWED
                                        ;
                                        ;-

011746                                  .RCOST:
011746                                          GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK

011766  016767  000000G 000064G                 MOV     .PNUMB,$BFR+D$MAXC      ; STORE THE MAXIMUM COST FOR NETWORK
011774  001404                                  BEQ     10$                     ; IF ZERO, ILLEGAL
011776  022767  001777  000000G                 CMP     #MAXCST,.PNUMB          ; IS THE NUMBER IN RANGE ?
012004  103003                                  BHIS    20$                     ; IF HIS, YES
012006  062716  000002          10$:            ADD     #2,(SP)                 ; ELSE, REJECT TRANSITION
012012                                          RETURN

012014                          20$:            PUTRC                           ; WRITE BACK DECNET HOME BLOCK

012020                                          RETURN
```

```
   3617                                               .SBTTL  ACTION ROUTINE: VERIFICATION LEVEL SEEN
   3618                                       ;+
   3619                                       ; *** - .OVFY - VERIFICATION LEVEL SEEN
   3620                                       ;
   3621                                       ; INPUTS:
   3622                                       ;       .PNUMB = THE VERIFICATION LEVEL
   3623                                       ;
   3624                                       ; OUTPUTS:
   3625                                       ;       TRANSITION IS ACCEPTED/REJECTED
   3626                                       ;
   3627                                       ;-
   3628
   3629 013356  116767  000000G 000062' .OVFY:: MOVB    .PNUMB,FLAG        ; SET VERIFICATION LEVEL
   3630 013364  022767  000010  000000G         CMP     #10,.PNUMB         ; IS VERIFICATION LEVEL IN RANGE ?
   3631 013372  101002                          BHI     10$                ; IF HI, YES
   3632 013374  062716  000002                  ADD     #2,(SP)            ; ELSE, REJECT TRANSITION
   3633 013400                          10$:    RETURN
```

```

                                      ;+
                                      ; PAS - SERVICE BLOCK PASSWORD
                                      ;
                                      ; INPUTS -
                                      ;          .PSTPT - ADDRESS OF PARSED STRING
                                      ;          .PSTCN - SIZE OF STRING
                                      ;
                                      ; OUTPUTS -
                                      ;          SERPAS - CONTAINS SERVICE BLOCK PASSWORD
                                      ;-

015204  010446                        PAS::     MOV     R4,-(SP)         ; SAVE REG
015206  010346                                  MOV     R3,-(SP)         ; SAVE TPARS REG
015210  012700  000164'                         MOV     #HEXSTR,R0       ; GET BUFFER ADDRESS
015214  112720  000060                1$:       MOVB    #'0,(R0)+        ; FILL IN DEFAULT VALUES
015220  020027  000204'                         CMP     R0,#ENDHEX       ; END OF BUFFER ?
015224  001373                                  BNE     1$               ; IF NOT - BRANCH

015226  012700  000164'                         MOV     #HEXSTR,R0       ; GET TEMPORARY BUFFER ADDRESS
015232  066700  000000G                         ADD     .PSTCN,R0        ; STORE STRING IN REVERSE ORDER

015236  016701  000000G                         MOV     .PSTCN,R1        ; GET SIZE OF STRING
015242  016702  000000G                         MOV     .PSTPT,R2        ; GET STRING ADDRESS
015246  010104                                  MOV     R1,R4            ; SAVE STRING LENGTH

015250  112240                        10$:      MOVB    (R2)+,-(R0)      ; STORE STRING IN WORK BUFFER
015252  005301                                  DEC     R1               ; COUNT ONE LESS
015254  001375                                  BNE     10$              ; IF MORE - BRANCH
015256  012702  000122'                         MOV     #SERPAS,R2       ; GET PASSWORD BUFFER ADDRESS
015262  012700  000164'                         MOV     #HEXSTR,R0       ; SET START ADDRESS OF PASSWORD

015266                                20$:      CALL    HEXIND           ; CONVERT FROM HEX
015272  110312                                  MOVB    R3,(R2)          ; STORE RESULT IN PASSWORD BUFFER
015274                                          CALL    HEXIND           ; CONVERT FROM HEX
015300  000241                                  CLC                      ; CLEAR C-BIT

015302  006303                                  ASL     R3               ; POSITION VALUE
015304  006303                                  ASL     R3               ; ...
015306  006303                                  ASL     R3               ; ...
015310  006303                                  ASL     R3               ; ...

015312  150322                                  BISB    R3,(R2)+         ; STORE RESULT IN PASSWORD BUFFER
015314  022702  000132'                         CMP     #ENDPAS,R2       ; CHECK FOR LAST CHARACTER
015320  001362                                  BNE     20$              ; IF MORE - BRANCH
015322  012603                                  MOV     (SP)+,R3         ; RESTORE REG
015324  012604                                  MOV     (SP)+,R4         ; RESTORE REG

015326                                          RETURN
```

```
 4589 017010  016700  000332'           DEV::   MOV     STRAD,R0         ; GET STRING ADDRESS
 4590 017014  016701  000204'                   MOV     STRSIZ,R1        ; GET STRING SIZE
 4591 017020  110167  000242'                   MOVB    R1,DEVBUF        ; STORE STRING SIZE
 4592 017024  012702  000243'                   MOV     #DEVBUF+1,R2     ; GET STARTING ADDRESS OF STRING
 4593
 4594 017030  112022                    1$:     MOVB    (R0)+,(R2)+      ; STORE BYTE
 4595 017032  005301                            DEC     R1               ; COUNT ONE LESS
 4596 017034  001375                            BNE     1$               ; IF MORE - BRANCH
 4597 017036  112712  000072                    MOVB    #':,(R2)         ; STORE COLON
 4598 017042  105267  000242'                   INCB    DEVBUF           ; COUNT THE :
 4599 017046  012767  000266' 000334'           MOV     #FNBUFF,STRPT    ; RESET START OF BUFFER ADDRESS
 4600 017054  005067  000204'                   CLR     STRSIZ           ; INIT COUNT
 4601
 4602 017060                                    RETURN
 4603
```

```
   1494 006514                                     RETURN                                                         ;[TD001]
   1495                                    .                                                                      ;[TD001]
   1496 006516   000261               99$:         SEC                             ; Signal string too long       ;[TD001]
   1497 006520                                     RETURN                                                         ;[TD001]
   1498
```

```
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE             REFERENCES

OP$SCO  = 000001          #5-61
OP$SEG  = 000400          #5-61
OP$SER  = 000400          #5-61
OP$SIN  = 000010          #5-61
OP$STA  = 000004          #5-61
OP$TPA  = 000020          #5-61
OP$TRI  = 000040          #5-61
OP$UCS  = 000020          #5-61
OP$USE  = 000002          #5-61
OP$VE(  = 000040          #5-61
OP$VER  = 000040          #5-61
OP$XAC  = 000001          #5-61      7-596
OP$XPR  = 000002          #5-61      7-600
OP$X5S  = 000004          #5-61      7-604
OP$X9S  = 000010          #5-61      7-608
OTSKTL    001347 R        #5-168     5-168      7-809
PARTAB  = ******   GX      6-212
PASWTL    000637 R        #5-158     5-158      7-1148     7-1185
PCKBCD    006204 R         7-507    #9-1341
PF.XDF  = ******   GX      13-1476
PNAMTL    000172 R        #5-148     5-148      7-869
PRNMTL    000321 R        #5-151     5-151      7-1127
PSWFTL    001604 R        #5-173     5-173      7-490
QF$ACT  = 000001          #5-61
QF$ALL  = 000002          #5-61
QF$KNO  = 000004          #5-61
QF$LOO  = 000010          #5-61
Q.IOFN  = 000002           5-106     5-112
Q.IOPL  = 000014           5-107     5-113
RAD       006052 R         7-636     7-804      7-838      7-865      7-1122    #8-1286
RQB     = ******   GX      6-204     7-263      7-275      7-279      7-309     *7-346     *7-407     *7-411     *7-422
                          *7-436    *7-452     *7-454      7-466      7-483      7-506     *7-513      7-536     *7-551
                           7-559    *7-572     *7-640     *7-642     *7-653     *7-664     *7-676     *7-687     *7-698
                          *7-709    *7-720     *7-732     *7-744     *7-806     *7-807     *7-819     *7-820     *7-821
                          *7-822    *7-823     *7-831     *7-840      7-850     *7-867     *7-878     *7-911     *7-924
                          *7-938    *7-965     *7-966      7-975      7-983      7-992     *7-994     *7-1017    *7-1018
                           7-1028   *7-1030     7-1048    *7-1050    *7-1054    *7-1078    *7-1079     7-1096    *7-1124
                          *7-1125    7-1132     7-1159     7-1169    *7-1199    *7-1201    *7-1219    *7-1235    *7-1237
                          *7-1260  *10-1368    11-1404    12-1428    12-1430
RQBST   = ******   GX     *6-248     7-617
R$$EIS  = ******           9-1355
R$$11D  = ******           9-1355
SCODV     000060 R        #5-116    *7-887     *7-890      7-897
SCOUN     000062 R        #5-117     5-118     *7-893      7-900
SPACE   = 000040          #5-79      15-1537
SPFILL    006536 R         7-278     13-1473   #15-1533
STORID    006416 R         7-265     7-561      7-852    #13-1468
STRNAM    004474 R         7-993     7-1029    #7-1034     7-1049
SYNERR    000114 R        #5-146     5-146      6-243
SYSFDB  = ******   GX     *7-549    *7-780     *7-781     *7-782     *7-952     *7-953     *7-954     *7-1003    *7-1004
                          *7-1005   *7-1064    *7-1065    *7-1066    *8-1317

TRIBTL    000737 R        #5-160     5-160      7-1208
```

```
                                         .SBTTL  PROC - PROCESS SPECIAL REQUESTS
                                ;+
                                ; **-PROC-PROCESS SPECIAL REQUESTS
                                ;
                                ; INPUTS:
                                ;        RQB = FUNCTION REQUST BLOCK
                                ;
                                ; OUTPUTS:
                                ;        NONE.
                                ;
                                ;-

000134                          PROC:
000134  116700  000000G                 MOVB    RQB,R0                  ; PICK UP FUNCTION CODE
000140  012701  000164'                 MOV     #PRTBL,R1               ; POINT TO SPECIAL PROCESSING TABLE
000144  012102                  10$:    MOV     (R1)+,R2                ; GET FUNCTION CODE TO TEST
000146  001540                          BEQ     $VRQB                   ; IF TABLE END, NOTHING SPECIAL TO DO
000150  020200                          CMP     R2,R0                   ; IS THIS THE CORRECT FUNCION?
000152  001402                          BEQ     20$                     ; BR IF YES - DO SPECIAL PROCESSING
000154  005721                          TST     (R1)+                   ; POINT PAST PROCESSING ROUTINE
000156  000772                          BR      10$                     ; AND CONTINUE
000160  000171  000000          20$:    JMP     @(R1)                   ; ELSE TAKE CARE OF SPECIAL CASES

                                ;
                                ; SPECIAL PROCESSING DISPATCH TABLE
                                ;
000164  000003  000202'         PRTBL:  .WORD   LS.LIN,PRLIN            ; SET LINE
000170  000006  000202'                 .WORD   LX.LIN,PRLIN            ; CLEAR LINE
000174  000033  000652'                 .WORD   LS.CIR,PRCIR            ; SET CIRCUIT
000200  000000                          .WORD   0                       ; END OF TABLE
```

```
    622                                                         .SBTTL  $SCIR - SET CIRCUIT
    623                                                 ;+
    624                                                 ; $SCIR - "SET CIRCUIT"
    625                                                 ;
    626                                                 ; INPUTS:
    627                                                 ;       $OPTON = OPTIONS WORD
    628                                                 ;
    629                                                 ; OUTPUTS:
    630                                                 ;       C-BIT=SUCCESS/FAILURE
    631                                                 ;
    632                                                 ;-
    633
    634 002024                                          $SCIR:
    635 002024  005767  000000G                                 TST     $OPTON                  ; INITIAL CIRCUIT LOAD?
    636 002030  001417                                          BEQ     30$                     ; BR IF YES
    637 002032  012705  002174'                                 MOV     #CIRTAB,R5              ; ELSE GET FUNCTION DISPATCH TABLE
    638 002036  012504                                  10$:    MOV     (R5)+,R4                ; GET OPTIONS BIT TO CHECK
    639 002040  001454                                          BEQ     60$                     ; BR IF END OF TABLE
    640 002042  030467  000000G                                 BIT     R4,$OPTON               ; WAS THIS OPTION SPECIFIED?
    641 002046  001406                                          BEQ     20$                     ; BR IF NO
    642 002050  010446                                          MOV     R4,-(SP)                ; SAVE R4
    643 002052  010546                                          MOV     R5,-(SP)                ; SAVE R5
    644 002054                                                  CALL    @(R5)                   ; ELSE, EXECUTE REQUEST
    645 002060  012605                                          MOV     (SP)+,R5                ; RESTORE R5
    646 002062  012604                                          MOV     (SP)+,R4                ; RESTORE R4
    647 002064  005725                                  20$:    TST     (R5)+                   ; SKIP TO NEXT REQUEST
    648 002066  000763                                          BR      10$                     ; IF NE, YES
    649
    650 002070  016701  000000G                         30$:    MOV     $DDMPV,R1               ; GET PDV INDEX
    651 002074                                                  CALL    AIC                     ; IS PROCESS ALREADY LOADED?
    652 002100  001014                                          BNE     40$                     ; BR IF YES
    653 002102  012767  000000G 000000G                         MOV     #MS.INC!MS.EXT,$MISS    ; LOOK FOR PROCESS INCREMENT
    654 002110                                                  CALL    $NLCFG                  ; ...
    655 002114  103426                                          BCS     60$                     ; IF CS, FAILED
    656 002116                                                  CALL    $XFER                   ; TRANSFER DDM INTO CORE
    657 002122  103423                                          BCS     60$                     ; IF CS, FAILURE (NOTHING TO CLEAN UP)
    658 002124  052767  000000G 000000G                         BIS     #FL.PX2,$FLAGS          ; INDICATE SUCCESSFUL DDM TRANSFER
    659 002132  016700  000000G                         40$:    MOV     $SLTA,R0                ; GET SLT ADDRESS
    660 002136  012767  000000G 000000G                         MOV     #MS.CIR,$MISS           ; INDICATE CIRCUIT PARAM. ARE MISSING
    661 002144                                                  CALL    $NLCFG                  ; SCAN CETAB
    662 002150  103410                                          BCS     60$
    663 002152  016700  000000G                                 MOV     $SLTA,R0                ; GET SLT ADDRESS
    664 002156  032760  000000G 000000G                         BIT     #LF.RDY,L.FLG(R0)       ; HAS LINE TABLE BEEN LOADED?
    665 002164  001002                                          BNE     60$                     ; BR IF YES
    666 002166                                                  CALL    $TMP                    ; ELSE EXPAND LINE TABLE
    667
    668 002172                                          60$:    RETURN
    669
    670 002174                                          CIRTAB:
    671 002174  000001  000000G                                 .WORD   OP$LTM,$VLTM            ; SET LISTEN TIMER
    672 002200  000002  000000G                                 .WORD   OP$HTM,$VHTM            ; SET HELLO TIMER
    673 002204  000004  000000G                                 .WORD   OP$STA,$VENA            ; ENABLE CIRCUIT
    674 002210  000010  000000G                                 .WORD   OP$MAC,$VACT            ; SET MULTIPOINT ACTIVE RATIO
    675 002214  000020  000000G                                 .WORD   OP$OWN,$VOWN            ; SET CIRCUIT OWNER
    676 002220  000040  000000G                                 .WORD   OP$TRI,$VTAD            ; SET TRIBUTARY ADDRESS
    677 002224  000100  000000G                                 .WORD   OP$COS,$VCOST           ; SET CIRCUIT COST
    678 002230  000400  000000G                                 .WORD   OP$SER,$VSER            ; SET CIRCUIT SERVICE
```

```
                                                                    C  5
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45     PAGE 6

SYMBOL CROSS REFERENCE                                  CREF   04.00

SYMBOL  VALUE            REFERENCES

$DSYS   = ******  GX      13-454
$EROUT  = ******  GX      10-261     10-262     11-306     12-390     15-521     15-533
$FLAGS  = ******  GX     *16-575     16-579    *17-658    *19-743    *19-754    *20-782    *20-783
$FUNC   = ******  GX      11-282     11-291    *11-297     11-298
$GETRV  = ******  GX      20-790
$MISS   = ******  GX      11-284    *17-653    *17-660    *19-738    *19-749
$NAME   = ******  GX     *20-784
$NLCFG  = ******  GX      11-286     17-654     17-661     19-739     19-750
$NLDEV  = ******  GX      16-585
$NLRQB  = ******  GX      11-280
$OPTON  = ******  GX      10-211     10-216    *10-249     12-334    *12-378     17-635     17-640     18-694     18-701
$PANIC    000130 RG       7-144     #8-159      13-443
$PDVA   = ******  GX      16-570     16-576     16-588     16-611    *20-779
$PDVTA  = ******  GX      14-489
$QUAL   = ******  GX      10-206     10-208     10-219     10-227     10-234     10-253     12-326     12-328     12-337
                          12-345     12-352     12-382
$RADDR  = ******  GX     *20-790
$ROL    = ******  GX      16-592
$RSNTI  = ******  GX      11-301
$SAVAL  = ******  GX      11-277     15-511
$SCEX   = ******  GX      13-435
$SCIR     002024 R        13-461    #17-634
$SLIN     002236 R        13-437    #18-693
$SLTA   = ******  GX      17-659     17-663
$SLTMA  = ******  GX      10-214     12-332
$SLTNM  = ******  GX      10-215     12-333
$SMOD   = ******  GX      13-464
$SNOD   = ******  GX      13-446
$SPRO     001614 R        13-436    #16-569
$TMP    = ******  GX      16-581     17-666     19-744     19-760
$VACT   = ******  GX      17-674
$VALI   = ******  GX      13-448     13-449
$VCOST  = ******  GX      17-677
$VDEA   = ******  GX      18-713
$VENA   = ******  GX      17-673
$VFLAG  = ******  GX      11-288     11-299
$VHELP  = ******  GX      13-444
$VHTM   = ******  GX      17-672
$VINI   = ******  GX      7-143
$VLOG   = ******  GX      13-450     13-451
$VLTM   = ******  GX      17-671
$VNCP   = ******  GX      7-146
$VNP      000104 RG      #7-142      20-795
$VNTI   = ******  GX      13-445
$VOBJ   = ******  GX      13-452     13-453
$VOWN   = ******  GX      17-675
$VRQB     000450 RG       9-177      10-209     10-247    #11-276     12-329     12-373
$VSER   = ******  GX      17-678
$VTAD   = ******  GX      17-676
$VTST   = ******  GX      13-442
$VXIT   = ******  GX      8-159
$XFER   = ******  GX      16-573     17-656     19-741     19-752


                                                                    D  5
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45     PAGE 7
```

```
          .TITLE  VNTI - VNP NETWORK INITIALIZATION ROUTINES
          .IDENT  /V05.00/
;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;         VNP - NETWORK BUFFER POOL INITIALIZATION ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00    27-FEB-78
;         VERSION 2.0 RELEASE
;
; 2.00    14-DEC-79
;         DECNET-11M/S V3.0
;         DECNET-11M-PLUS V1.0
;
; 3.00    16-APR-82
;         DECNET-11M V3.1
;         DECNET-11M-PLUS V1.1
;
; 4.00    07-NOV-83
;         DECNET-11M V4.0
;         DECNET-11M-PLUS V2.0
;
; 5.00    22-JUL-85
;         DECnet-11M/S V4.2
;         DECnet-11M-Plus V3.0
;         DECnet-Micro/RSX V1.0
;
;

```

```
        001771     052     040     055
        001774     055     040
   615          000024                          PREWLN=.-PREWRN
   616                                  ;
   617                                  ; SCRATCH BUFFER
   618                                  ;
   619  001776                          SBUF:   .BLKB   8.
   620                                          .EVEN
   621                                  ;
   622                                  ; OUTPUT QIO BLOCK
   623                                  ;
   624  002006                          QIO:    QIOW$   IO.WVB,2,2,,,,<0,0,40>
   625                                  ;
   626                                  ; ERROR MESSAGES
   627                                  ;
   628                                          .NLIST  BEX
   629  002036                          ERRTAB: ERRDEF  1,<KMC microcode load failed>
   630  002042                                  ERRDEF  2,<Unable to dynamically allocate network buffer pool>
   631  002046                                  ERRDEF  3,<Network pool partition not in the system>
   632  002052                                  ERRDEF  4,<Communications executive load failed>
   633  002056                                  ERRDEF  5,<Syntax error in PAR$DF macro>
   634  002062                                  ERRDEF  6,<Unspecified error during config file scan>
   635  002066                                  ERRDEF  7,<Resource error trying to allocate CCBs>
   636  002072                                  ERRDEF  10,<Network buffer pool is too small>
   637  002076                                  ERRDEF  11,<Resource error trying to allocate RDBs>
   638  002102                                  ERRDEF  12,<Resource error trying to allocate SDBs>
   639  002106                                  ERRDEF  13,<Configuration file read error>
   640  002112                                  ERRDEF  14,<Open failure on configuration file>
   641  002116                                  ERRDEF  15,<Syntax error in BUF$DF macro>
   642  002122                                  ERRDEF  16,<Failed to load a system line>
   643  002126                                  ERRDEF  17,<Failed to load a process>
   644  002132                                  ERRDEF  20,<UMR block allocation failure>
   645  002136                                  ERRDEF  21,<Not enough contiguous UMRs to map network pool>
   646  002142                                  ERRDEF  22,<Non-UMR-mapped area is too large>
   647  002146                                  ERRDEF  23,<Object!Remote!Filter block allocation failure>
   648  002152                                  ERRDEF  24,<Syntax error in ROU$DF macro>
   649  002156                                  ERRDEF  25,<Dummy message for postion>
   650  002162                                  ERRDEF  26,<Line not in system>
   651  002166                                  ERRDEF  27,<Illegal counter timer value>
   652  002172                                  ERRDEF  30,<Same as 25>
   653  002176                                  ERRDEF  31,<Failed to load a DLM circuit>
   654  002202                                  ERRDEF  32,<Filename block allocation failure>
   655  002206                                  ERRDEF  33,<Sevice block allocation failure>
   656  002212                                  ERRDEF  34,<Illegal max logical links value>
   657  002216                                  ERRDEF  35,<Illegal max node counter value>
   658  002222                                  ERRDEF  36,<Illegal delay factor value>
   659  002226                                  ERRDEF  37,<Illegal delay weight value>
   660  002232                                  ERRDEF  40,<Illegal packet limiter value>
   661  002236                                  ERRDEF  41,<Illegal link service threshold value>
   662  002242                                  ERRDEF  42,<Illegal incoming timer value>
   663  002246                                  ERRDEF  43,<Illegal outgoing timer value>
   664  002252                                  ERRDEF  44,<Illegal inactivity timer value>
   665  002256                                  ERRDEF  45,<Illegal retransmit factor value>
   666
   667                                          .EVEN
   668                                          .LIST BEX
```

```
                .SBTTL  STATE TRANSITIONS FOR FINDING NETWORK POOL PARTITION

003632          ISTAT$  IBUF2,KBUF2

003632          STATE$
003632           TRAN$   %PAR$DF%,,,1,CFSFLG     ; SIGNAL THAT MACRO WAS FOUND
003632          STATE$
003632           TRAN$   <',>,X1
003632           TRAN$   $RAD50,,.NAM1           ; FIRST PARTITION NAME
003632          STATE$
003632           TRAN$   <',>
003632          STATE$  X1
003632           TRAN$   <',>,X2
003632           TRAN$   $RAD50,,.NAM2           ; SECOND PARTITION NAME
003632          STATE$
003632           TRAN$   <',>
003632          STATE$  X2
003632           TRAN$   $NUMBR,,.FUDG           ; FUDGE FACTOR
003632          STATE$
003632           TRAN$   $EOS,$EXIT
003632           TRAN$   <',>
003632          STATE$
003632           TRAN$   <',>,X3
003632           TRAN$   "TOP",,,1,TOPFLG        ; REQUEST TOP-LOADING
003632          STATE$
003632           TRAN$   $EOS,$EXIT
003632           TRAN$   <',>
003632          STATE$  X3
003632           TRAN$   $NUMBR,X4,.NOMAP        ; SIZE OF NON-UMR-MAPPED AREA
003632           TRAN$   "NONE",,,1,UMRFLG       ; DISABLE UMR ALLOCATION
003632          STATE$  X4
003632           TRAN$   $EOS,$EXIT
003632           TRAN$   <',>
003632          STATE$
003632           TRAN$   $NUMBR,,.ALLC           ; SIZE OF SINGLE-WORD POOL
003632          STATE$
003632           TRAN$  $EOS,$EXIT
003632           TRAN$  <',>
003632          STATE$
003632           TRAN$  %SBPOOL%,$EXIT,,1,SBFLG ; SERVER BASE FLAG
003632          STATE$

```

```
                                               .SBTTL  PERFORM CONFIGURATION FILE SCAN
                                       ;+
                                       ;       **-$CFG-PERFORM CONFIGURATION FILE SCAN
                                       ;
                                       ;       THE CONFIGURATION FILE CONTAINS INFORMATION ABOUT THE NETWORK
                                       ;       WHICH WE WILL WANT TO LOAD. THE USER MAY HAVE EDITED THE
                                       ;       CONFIGURATION FILE IN ORDER TO CHANGE DYNAMIC SYSTEM VARIABLES.
                                       ;-
                                       ;       OUTPUTS:
                                       ;               'C' CLEAR - CONFIGURATION FILE SCAN SUCCESSFUL
                                       ;               'C' SET   - ERROR SCANNING CONFIGURATION FILE
                                       ;
005410                                 $CFG::  CALL    $SAVAL                  ; SAVE ALL REGISTERS ON ENTRY
005414                                         CALL    $CLSAV                  ; SAVE GCL CONTEXT
005420  112767  000100  000000G                MOVB    #CL.OPN,$CLFLG          ; LEAVE FILE OPEN
005426  016701  000000G                        MOV     $PDVTA,R1               ; POINT TO THE PDV INDEX TABLE
005432  016700  000000G                        MOV     $PDVNM,R0               ; NUMBER OF PDV ENTRIES
005436  012102                         10$:    MOV     (R1)+,R2                ; GET ADDRESS OF PDV
005440  142762  000010  000010                 BICB    #ZF.MFL,Z.FLG(R2)
005446  005300                                 DEC     R0
005450  001372                                 BNE     10$                     ; LOOP TILL DONE
005452  016702  000000G                        MOV     $SLTMA,R2               ; POINT TO SLT MAPPING TABLE
005456  016700  000000G                        MOV     $SLTNM,R0               ; NUMBER OF SYSTEM LINES
005462  001406                                 BEQ     30$                     ; IF EQ, NONE
005464  012201                         20$:    MOV     (R2)+,R1                ; GET NEXT SYSTEM LINE TABLE
005466  042761  006040  000000                 BIC     #LF.MFL!LF.ENA!LF.SER,L.FLG(R1) ; CLEAR DYNAMIC BITS
005474  005300                                 DEC     R0
005476  003372                                 BGT     20$                     ; LOOP TILL DONE

005500  012767  177777  000000G 30$:           MOV     #-1,SLTNUM              ; INITIALISE SYSTEM LINE TABLE NUMBER
005506                                         CALL    OPNCFG                  ; ATTEMPT TO OPEN CONFIG FILE
005512  103421                                 BCS     100$                    ; FAILED TO OPEN CONFIG FILE
005514                                 40$:    GCL$                            ; GET A RECORD FROM THE FILE
005520  103411                                 BCS     50$                     ; END OF FILE REACHED?
005522  005067  000040'                        CLR     MRKFL                   ; INITIALIZE MARK FOR LOAD FLAG
005526                                         CALL    PROTXT                  ; NO ... PROCESS THE TEXT
005532  103370                                 BCC     40$                     ; LOOP UNLESS SYNTAX ERROR
005534                                         CALL    $CLDEQ                  ; DEQUE FILE ON SYNTAX ERROR
005540  000261                         45$:    SEC                             ; SIGNAL ERROR
005542                                         RETURN
005544  122777  000000G 000000G 50$:           CMPB    #IE.EOF,@$CLIOS         ; EOF ?
005552  001372                                 BNE     45$                     ; IF NOT, THEN ERROR ..
005554  000241                                 CLC                             ; SIGNAL SUCCESS
005556                                 100$:   RETURN                          ; RETURN WITH C-BIT CLEAR
```

```
                                        .SBTTL  OBJ$DF PARSE TABLES
                                ;+
                                ;       STATE TRANSITIONS FOR THE 'OBJ$DF' MACRO
                                ;-

006324                                  STATE$  OBJDF
006324                                   TRAN$  $NUMBR,,.ONUM

006324                                  STATE$
006324                                   TRAN$  <',>

006324                                  STATE$
006324                                   TRAN$  <',>,OBJ2
006324                                   TRAN$  LBRA

006324                                  STATE$
006324                                   TRAN$  $RAD50,,.ONAME

006324                                  STATE$
006324                                   TRAN$  RBRA

006324                                  STATE$
006324                                   TRAN$  <',>

006324                                  STATE$  OBJ2
006324                                   TRAN$  $NUMBR,,.OVFY

006324                                  STATE$
006324                                   TRAN$  <',>

006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.OFLAG

006324                                  STATE$
006324                                   TRAN$  !END,$EXIT,.OEND
006324                                   TRAN$  <',>

006324                                  STATE$
006324                                   TRAN$  $NUMBR,$EXIT,.OCPY
```

```
   2518 006324                                        STATE$  ZAS
   2519 006324                                         TRAN$  $LAMDA,,SERSLN
   2520
   2521 006324                                        STATE$
   2522 006324                                         TRAN$  <',>,$EXIT
   2523
   2524 006324                                        STATE$
   2525
   2526
```

```
                                                .SBTTL  ACTION ROUTINE: NODE NUMBER SEEN
                                        ;+
                                        ; *** - .NODNB - NODE NUMBER SEEN
                                        ;
                                        ; INPUT:
                                        ;       .PNUMB = THE NODE NUMBER
                                        ;
                                        ;-

 007236                                 .NODNB:
 007236                                         GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK

 007256 022767  001777  000000G                 CMP     #1023.,.PNUMB           ; CHECK LIMIT
 007264 002404                                  BLT     5$                      ; IF ILLEGAL - BRANCH
 007266 056767  000000G 000014G                 BIS     .PNUMB,$BFR+D$LNUM      ; STORE THE NODE NUMBER
 007274 001003                                  BNE     10$                     ; IF EQ, ILLEGAL NODE NUMBER
 007276 062716  000002                  5$:     ADD     #2,(SP)                 ; ELSE, REJECT TRANSITION
 007302                                         RETURN

 007304                                 10$:    PUTRC                           ; WRITE BACK DECNET HOME BLOCK
 007310                                         RETURN


                                        ;+
                                        ; *** - .NODAR - NODE AREA
                                        ;-
 007312                                 .NODAR: GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK

 007332 022767  000077  000000G                 CMP     #63.,.PNUMB             ; CHECK LIMIT
 007340 002423                                  BLT     10$                     ; IF ILLEGAL - BRANCH

 007342 016700  000000G                         MOV     .PNUMB,R0               ; COPY AREA
        000012                                  .REPT   10.
                                                ASL     R0                      ; POSITION AREA BITS
                                                .ENDR
 007372 000241                                  CLC
 007374 010067  000014G                         MOV     R0,$BFR+D$LNUM  ; STORE AREA NUMBER
 007400 001403                                  BEQ     10$                     ; IF ERROR - BRANCH

 007402                                         PUTRC                           ; WRITE BACK DECNET HOME BLOCK

 007406                                         RETURN

 007410 062716  000002                  10$:    ADD     #2,(SP)                 ; ILLEGAL AREA NUMBER
 007414                                         RETURN
```

```
                                                          .SBTTL  ACTION ROUTINE: MAXIMUM NUMBER OF HOPS
                                                 ;+
                                                 ; *** - .RHOP - MAXIMUM NUMBER OF HOPS IN NETWORK
                                                 ;
                                                 ; INPUT:
                                                 ;        .PNUMB = MAXIMUM NUMBER OF HOPS IN NETWORK
                                                 ;
                                                 ;-

012022                                           .RHOP:
012022                                                    GETRV   $DECPT,#D$END            ; READ DECNET HOME BLOCK

012042  016767  000000G 000066G                           MOV     .PNUMB,$BFR+D$MAXH       ; STORE THE MAXIMUM NUMBER OF HOPS
012050  016767  000000G 000070G                           MOV     .PNUMB,$BFR+D$MAXV       ; SET UP MAXIMUM NUMBER OF VISITS
012056  006367  000070G                                   ASL     $BFR+D$MAXV              ; MAXV = 2 * MAXH
012062  001404                                            BEQ     10$                      ; IF ZERO, ILLEGAL
012064  022767  000037  000000G                           CMP     #MAXHOP,.PNUMB           ; IS THE NUBMER IN RANGE ?
012072  103003                                            BHIS    20$                      ; IF HIS, YES
012074  062716  000002                           10$:     ADD     #2,(SP)                  ; ELSE, REJECT THE TRANSITION
012100                                                    RETURN

012102                                           20$:     PUTRC                            ; WRITE BACK DECNET HOME BLOCK

012106                                                    RETURN
```

```
                                                .SBTTL  ACTION ROUTINE: OBJECT FLAGS SEEN
                                        ;+
                                        ; *** - ACTION ROUTINE: OBJECT FLAGS SEEN
                                        ;
                                        ; INPUTS:
                                        ;       .PNUMB = OBJECT FLAGS
                                        ;
                                        ; OUTPUTS:
                                        ;       TRANSITION IS ACCEPTED/REJECTED
                                        ;
                                        ;-

 013402  156767  000000G 000062' .OFLAG::BISB   .PNUMB,FLAG             ; SETUP OBJECT FLAGS
 013410  032767  000007  000000G        BIT     #7,.PNUMB               ; OVER-WRITING VERIFICATION LEVEL ?
 013416  001003                         BNE     10$                     ; IF NE, YES
 013420  105767  000001G                TSTB    .PNUMB+1                ; IS IT A BYTE VALUE ?
 013424  001402                         BEQ     20$                     ; IF EQ, YES
 013426  062716  000002         10$:    ADD     #2,(SP)                 ; ELSE, REJECT TRANSITION
 013432                         20$:    RETURN
```

```

                                  ;+
                                  ; HEXIND - CONVERT FROM HEX
                                  ;
                                  ;-

 015330  005003                   HEXIND: CLR     R3              ; INIT REG
 015332  112003                           MOVB    (R0)+,R3        ; GET CHARACTER
 015334  032703  000100                   BIT     #LET,R3         ; CHECK FOR LETTER
 015340  001003                           BNE     10$             ; IF YES - BRANCH
 015342  162703  000060                   SUB     #'0,R3          ; CONVERT TO BINARY VALUE
 015346  000402                           BR      20$             ; CONTINUE
 015350  162703  000067           10$:    SUB     #<'A-10.>,R3    ; CONVERT HEX DIGIT TO BINARY
 015354                           20$:    RETURN


                                  ;+
                                  ; SERCR - SET UP CIRCUIT NAME
                                  ;
                                  ; INPUTS -
                                  ;       .PSTPT - PARSED STRING ADDRESS
                                  ;       .PSTCN - PARSED STRING SIZE
                                  ;
                                  ; OUTPUTS -
                                  ;       CIRNAM - CONTAINS DEVICE NAME (ASCII)
                                  ;-

 015356  012700  000150'          SERCR::MOV      #CIRNAM,R0      ; GET STRING STORAGE BUFFER
 015362  016702  000000G                  MOV     .PSTPT,R2       ; GET PARSED STRING ADDRESS
 015366  016701  000000G                  MOV     .PSTCN,R1       ; GET STRING COUNT
 015372  112220                   10$:    MOVB    (R2)+,(R0)+     ; STORE DEVICE NAME
 015374  005301                           DEC     R1              ; ANY LEFT ?
 015376  001375                           BNE     10$             ; IF YES - BRANCH

 015400                                   RETURN

                                  ;+
                                  ; FPDV - FIND PDV
                                  ;
                                  ; INPUTS -
                                  ;       R1 - RAD50 PROCESS NAME
                                  ;
                                  ; OUTPUTS -
                                  ;       R0 - PDV ADDRESS
                                  ;       R1- PDV INDEX
                                  ;-

 015402  010146                   FPDV:   MOV     R1,-(SP)        ; STORE NAME TO BE MATCHED
 015404  016701  000000G                  MOV     $PDVTA,R1       ; GET PDV VECTOR ADDRESS
 015410  016702  000000G                  MOV     $PDVNM,R2       ; GET NUMBER OF PDV'S
 015414  012100                   10$:    MOV     (R1)+,R0        ; GET NEXT PDV ADDRESS
 015416  026016  000004                   CMP     Z.NAM(R0),(SP)  ; MATCH ?
 015422  001405                           BEQ     40$             ; IF YES - BRANCH
 015424  005302                           DEC     R2              ; COUNT ONE LESS PDV
 015426  001372                           BNE     10$             ; IF MORE - BRANCH
 015430  005726                           TST     (SP)+           ; IF NO MORE - ERROR
 015432  000261                           SEC                     ; SET C-BIT
 015434  000404                           BR      50$             ; REPORT ERROR
```

```

                                        ;+
                                        ; GRPUIC - STORE GROUP CODE IN UICBUF
                                        ;
                                        ; INPUTS -
                                        ;       STRAD - PARSED STRING ADDRESS
                                        ;       STRSIZ - STRING SIZE
                                        ;
                                        ; OUTPUTS -
                                        ;       GROUP STRING STORED IN UICBUF
                                        ;
                                        ;-

 017062 016701  000204'                 GRPUIC::MOV     STRSIZ,R1       ; GET STRING SIZE
 017066 016700  000332'                         MOV     STRAD,R0        ; GET STRING ADDRESS

 017072 110167  000206'                         MOVB    R1,UICBUF       ; STORE GROUP UIC STRING SIZE
 017076 012702  000207'                         MOV     #UICBUF+1,R2    ; GET POINTER TO BUFFER
 017102 112722  000133                          MOVB    #'[,(R2)+       ; STORE BRACKET
 017106 105267  000206'                         INCB    UICBUF          ; COUNT BRACKET

 017112 112022                          1$:     MOVB    (R0)+,(R2)+     ; STORE STRING
 017114 005301                                  DEC     R1              ; COUNT ONE LESS
 017116 001375                                  BNE     1$              ; IF MORE - BRANCH

 017120 012767  000266' 000334' 10$:    MOV     #FNBUFF,STRPT   ; RESET BUFFER POINTER
 017126 005067  000204'                         CLR     STRSIZ          ; INIT SIZE

 017132                                         RETURN


                                        ;+
                                        ; USRUIC - STORE USER CODE IN UICBUF
                                        ;
                                        ; INPUTS -
                                        ;       STRAD - PARSED STRING ADDRESS
                                        ;       STRSIZ - STRING SIZE
                                        ;
                                        ; OUTPUTS -
                                        ;       USER UIC CODE STORED IN UICBUF
                                        ;-
 017134 052767  000001  000000' USRUI1::BIS     #1,NAMED        ; SET NAMED DIRECTORY FLAG
 017142 016701  000204'         USRUIC::MOV     STRSIZ,R1       ; GET SIZE
 017146 016700  000332'                 MOV     STRAD,R0        ; GET ADDRESS

 017152 005767  000000'                 TST     NAMED           ; NAMED DIRECTORY ?
 017156 001405                          BEQ     3$              ; IF NO - BRANCH
 017160 112767  000133  000207'         MOVB    #'[,UICBUF+1    ; LOAD [ CHAR
 017166 105267  000206'                 INCB    UICBUF          ; COUNT BRACKET

 017172 012702  000206'         3$:     MOV     #UICBUF,R2      ; GET BUFFER ADDRESS
 017176 005046                          CLR     -(SP)           ; ...
 017200 116716  000206'                 MOVB    UICBUF,(SP)     ; GET OLD UIC SIZE
 017204 062602                          ADD     (SP)+,R2        ; ADJUST BUFFER POINTER

 017206 005202                          INC     R2              ; INCLUDE SIZE BYTE
 017210 032767  000001  000000'         BIT     #1,NAMED        ; NAMED DIRECTORY ?
```

```
                                 .SBTTL  MOVE - Move bytes                                              ;[TD001]
                         ;+                                                                             ;[TD001]
                         ; *** MOVE - Move bytes                                                        ;[TD001]
                         ;                                                                              ;[TD001]
                         ; INPUTS                                                                       ;[TD001]
                         ;        R0 - Source address                                                   ;[TD001]
                         ;        R1 - Number of bytes to move                                          ;[TD001]
                         ;        R2 - Target address                                                   ;[TD001]
                         ;                                                                              ;[TD001]
                         ; OUTPUTS                                                                      ;[TD001]
                         ;        R2 - Address of last byte moved + 1                                   ;[TD001]
                         ;        R0, R1 destroyed                                                      ;[TD001]
                         ;                                                                              ;[TD001]
                         ;-                                                                             ;[TD001]
006522                   MOVE:                                                                          ;[TD001]
006522  005701                   TST     R1                      ; Check for nothing to do              ;[TD001]
006524  001403                   BEQ     20$                                                            ;[TD001]
006526  112022           10$:    MOVB    (R0)+,(R2)+                                                    ;[TD001]
006530  005301                   DEC     R1                                                             ;[TD001]
006532  003375                   BGT     10$                                                            ;[TD001]
006534                   20$:    RETURN                                                                 ;[TD001]
```

```
SYMBOL CROSS REFERENCE                                      CREF   04.00

SYMBOL   VALUE                REFERENCES

TY$CHA   = 000400             #5-61
TY$EVE   = 001000             #5-61
TY$STA   = 002000             #5-61
TY$SUM   = 000000             #5-61
UIDTL      001547 R           #5-172      5-172      7-543
UNUMTL     000421 R           #5-153      5-153      7-1244
VF.ADD   = ******   GX         7-552      7-579      7-990      7-1026     7-1046
VF.EAD   = ******   GX         7-1031
VF.INI   = ******   GX         6-209
VF.SKA   = ******   GX         7-1051     7-1055
VF.SS    = ******   GX         6-209
WC.CNT   = 000400             #5-61
WC.EVE   = 004000             #5-61
WC.TRI   = 002000             #5-61
WC.UNT   = 001000             #5-61
ZF.MUX   = ******   GX         7-1241     7-1273
Z.FLG    = ******   GX         7-1241     7-1273
Z.NAM    = ******   GX         11-1404
$ADS       000322 RG          #7-263
$ANT       000362 RG          #7-272
$ARAD1     002746 RG          #7-752
$ARAD2     002762 RG          #7-756
$BFR     = ******   GX         7-551      7-782     *7-896      7-901      7-903      7-906      7-908      7-910      7-954
                               7-1005     7-1066     8-1314     8-1318
$CAT5    = ******   GX         8-1302     8-1305
$CKLOG     000734 RG          #7-383
$CLBUF   = ******   GX         6-224      6-226
$CLIOS   = ******   GX         6-249
$CMAND   = ******   GX        *6-201      7-848
$COP       001002 RG          #7-404
$COST      001056 RG          #7-418
$CSR       001126 RG          #7-430
$CTL       001204 RG          #7-443
$DACC      001300 RG          #7-461
$DECPT   = ******   GX         7-549      7-781      7-953      7-1004     7-1065
$DEVHD   = ******   GX         7-896
$DPAS      001356 RG          #7-478
$DUSE      001572 RG          #7-531
$EXNOD     001650 RG          #7-548
$FLAGS   = ******   GX         7-1267    *7-1269
$GETRV   = ******   GX         7-549      7-781      7-953      7-1004     7-1065     8-1317
$GRO       001706 RG          #7-559
$GTPLN   = ******   GX         6-213
$HLPAC     006276 RG          #10-1368
$HMBUF   = ******   GX         7-780      7-784      7-952      7-956      7-1003     7-1007     7-1064     7-1068
$HTM       001746 RG          #7-569
$IFALI     002050 RG          #7-593
$IFLOG     002042 RG          #7-590
$IFNNA     002004 RG          #7-579
$IFXAC     002056 RG          #7-596
$IFXPR     002070 RG          #7-600
$IFX5S     002102 RG          #7-604
```

```
    192                                                      .SBTTL  PRLIN - SPECIAL PROCESSING FOR LINES
    193                                                ;+
    194                                                ; **-PRLIN-SPECIAL PROCESSING FOR LINES
    195                                                ;
    196                                                ; INPUTS:
    197                                                ;       RQB = REQUEST BLOCK
    198                                                ;       $QUAL = SPECIAL QUALIFERS
    199                                                ;
    200                                                ; OUTPUTS:
    201                                                ;       NONE.
    202                                                ;
    203                                                ;-
    204
    205 000202                                         PRLIN:
    206 000202  032767  000004  000000G                        BIT     #QF$KNO,$QUAL           ; IS THIS FOR KNOWN LINES ?
    207 000210  001004                                         BNE     10$                     ; IF NE, YES
    208 000212  032767  001400  000000G                        BIT     #WC.CNT!WC.UNT,$QUAL    ; ELSE, ARE THERE ANY WILD CARDS ?
    209 000220  001513                                         BEQ     $VRQB                   ; IF EQ, NO .. NOTHING SPECIAL HERE
    210
    211 000222  032767  000044  000000G 10$:           BIT     #OP$VEC!OP$CCS,$OPTON   ; ARE ANY INVALID OPTIONS PRESENT ?
    212 000230  001077                                         BNE     101$                    ; IF NE, YES .. ERROR
    213 000232  005004                                         CLR     R4                      ; INITIALIZE SUCCESS FLAG
    214 000234  016705  000000G                                MOV     $SLTMA,R5               ; GET ADDRESS OF SLT MAPPING TABLE
    215 000240  016703  000000G                                MOV     $SLTNM,R3               ; GET NUMBER OF SLT'S
    216 000244  116746  000000G                                MOVB    $OPTON,-(SP)            ; SAVE OPTIONS WORD
    217 000250                                 15$:            CALL    NXTLIN                  ; ELSE, GET THE NEXT LINE
    218 000254  103455                                         BCS     60$                     ; IF CS, NO MORE
    219 000256  032767  000004  000000G                        BIT     #QF$KNO,$QUAL           ; IS THIS FOR KNOWN LINES ?
    220 000264  001024                                         BNE     30$                     ; IF NE, YES
    221
    222 000266  026167  000000G 000002G                        CMP     Z.NAM(R1),RQB+LR.LIN    ; IS THIS THE RIGHT LINE ?
    223 000274  001365                                         BNE     15$                     ; IF NE, NO
    224                                                ;
    225                                                ; MATCH ON CONTROLLER
    226                                                ;
    227 000276  032767  000400  000000G                        BIT     #WC.CNT,$QUAL           ; IS THIS FOR ALL CONTROLLERS ?
    228 000304  001004                                         BNE     20$                     ; IF NE, YES
    229 000306  126067  000000G 000004G                        CMPB    L.CTL(R0),RQB+LR.CTL    ; ELSE, IS THIS THE RIGHT CONTROLLER ?
    230 000314  001355                                         BNE     15$                     ; IF NE, NO
    231                                                ;
    232                                                ; MATCH ON UNIT NUMBER
    233                                                ;
    234 000316  032767  001000  000000G 20$:           BIT     #WC.UNT,$QUAL           ; IS THIS FOR ALL UNITS ?
    235 000324  001004                                         BNE     30$                     ; IF NE, YES
    236 000326  126067  000000G 000005G                        CMPB    L.UNT(R0),RQB+LR.UNT    ; ELSE, IS THIS THE RIGHT UNIT ?
    237 000334  001345                                         BNE     15$                     ; IF NE, NO
    238                                                ;
    239                                                ; SAVE DATA IN REQUEST BLOCK
    240                                                ;
    241 000336  005204                          30$:           INC     R4                      ; INDICATE A LINE FOUND
    242 000340  016167  000000G 000002G                        MOV     Z.NAM(R1),RQB+LR.LIN    ; SAVE DDM PROCESS NAME
    243 000346  116067  000000G 000004G                        MOVB    L.CTL(R0),RQB+LR.CTL    ; SAVE CONTROLLER NUMBER
    244 000354  116067  000000G 000005G                        MOVB    L.UNT(R0),RQB+LR.UNT    ; SAVE UNIT NUMBER
    245 000362  105067  000006G                                CLRB    RQB+LR.TRI              ; ASSUME LINE IS NOT MULTIPOINT
    246 000366  016746  000000G                 50$:           MOV     RQB,-(SP)               ; SAVE FUNCTION REQUESTED
    247 000372                                                 CALL    $VRQB                   ; PROCESS REQUEST
    248 000376  012667  000000G                                MOV     (SP)+,RQB               ; RESTORE FUNCTION CODE JUST IN CASE
    679 002234  000000                      .WORD   0
```

```
                                                              D  5
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 7
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE           REFERENCES

$XLIN   = ******  GX     19-774
$XPRO   = ******  GX     16-620
$$      = 177777         6-83       6-83      #6-83     #6-84       6-85       6-85     #6-85     #6-86       6-87
                         6-87      #6-87      #6-88      11-309     11-309    #11-309   #11-310    15-541     15-541
                        #15-541    #15-542     15-543    15-543    #15-543    #15-544
$$CSR   = ******  GX     15-544
$$DEV   = ******  GX     16-583
$$HTIM  = ******  GX     19-765
$$PRI   = ******  GX     15-525
$$VECT  = ******  GX     15-544
$$$     = 001500 R      #6-83       6-84      #6-85       6-86      #6-87       6-88     #11-309    11-310    #15-516
                         15-516    #15-520     15-520   #15-528     15-528    #15-532     15-532   #15-541     15-542
                        #15-543     15-544
$$$$    = 001546 R      #6-84      #6-86      #6-88     #11-310    #15-542    #15-544
.MGMGE  = ******  GX     20-788
.PLGTH  = ******  GX     20-790
```

```
                                                              E  5
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 8
```

```
                                          .SBTTL  MACRO CALLS

                                          .MCALL  GCLDF$,CCBDF$,PDVDF$,SLTDF$,NHWDF$,VNPDF$,ASR$,QIOW$,PCBDF$
                                          .MCALL  ISTAT$,STATE$,TRAN$,GETRV,GETRC,PUTRC,GETAD,PUTAD,SCBDF$,ASL$
                                          .MCALL  SWTCHI,SWTCHO,HWDDF$,OBJDF$,RNBDF$,FLTDF$,OPTDF$,CEACC$
                                          .MCALL  SAVRG,RESRG,DHBDF$,CTRDF$,LLTDF$,FNBDF$,SERDF$,ADJDF$,CALLR
                                          .ENABL  LC

 000000                                   OBJDF$                  ; DEFINE OBJECT BLOCK OFFSETS
 000000                                   ADJDF$                  ; DEFINE ADJACENCY DATA BASE OFFSETS
 000000                                   LLTDF$                  ; DEFINE LOGICAL LINK SIZE
 000000                                   RNBDF$                  ; DEFINE REMOTE BLOCK OFFSETS
 000000                                   OPTDF$                  ; DEFINE COMM/EXEC OPTIONS
 000000                                   DHBDF$                  ; DEFINE DECNET HOME BLOCK OFFSETS

                000001                    M$$MGE=1                ; FORCE MEMORY MANAGEMENT
 000000                                   CTRDF$                  ; DEFINE COUNTER BLOCK OFFSETS
 000000                                   GCLDF$                  ; GET COMMAND LINE OFFSETS
 000000                                   CCBDF$                  ; CCB OFFSETS
 000000                                   PDVDF$                  ; PROCESS DESCRIPTOR VECTOR OFFSETS
 000000                                   SLTDF$                  ; SYSTEM LINE TABLE OFFSETS
 000000                                   PCBDF$                  ; PARTITION CONTROL BLOCK OFFSETS
 000000                                   NHWDF$                  ; NETWORK HARDWARE DEFINITIONS
 000000                                   SCBDF$                  ; UMR BLOCK OFFSETS
 000000                                   FLTDF$                  ; DEFINE FILTER BLOCK OFFSETS AND SYMBOLS
 000000                                   VNPDF$                  ; VNP OFFSETS AND FLAG WORD VALUES
 000000                                   HWDDF$                  ; HARDWARE DEFS
 000000                                   FNBDF$                  ; FILE NAME BLOCK OFFSETS DEFINITIONS
 000000                                   SERDF$                  ; SERVICE BLOCK DEFINITIONS


                                  ;
                                  ; DEFINE MACROS USED FOR ERROR REPORTING
                                  ;

                                          .MACRO NTIERR   ERN
                                          MOV     #'ERN,-(SP)     ; STORE ERROR CODE ON STACK
                                          JMP     NTIERR          ;  AND JUMP TO ERROR PROCESSING ROUINE
                                          .ENDM   NTIERR

                                          .MACRO  NTWARN  ERN
                                          MOV     #'ERN,-(SP)     ; STORE ERROR CODE ON STACK
                                          CALL    NTWARN          ; PRINT WARNING MESSAGE
                                          TST     (SP)+           ; POP STACK
                                          .ENDM   NTWARN

                                          .MACRO  ERRDEF ERN,ERRTXT
                                          .PSECT  DATA,D          ; DATA PSECT
                                  ERR'ERN:.ASCII  "ERRTXT"
                                          E'ERN'LEN=.-ERR'ERN
                                          .EVEN
                                          .PSECT
                                          .WORD   ERR'ERN
                                          .WORD   E'ERN'LEN
                                          .ENDM   ERRDEF
```

```
                                                .SBTTL UMR ALLOCATION
                                        ;+
                                                **-$UMRAL       - ALLOCATE UMRS FOR NETWORK BUFFER POOL
                                        ;
                                        ; INPUTS:
                                                $CPBIA  - CPU LOWER BIAS  (MULTIPROCESSOR ONLY)                 ;[MP01]
                                                PSTART  - NETWORK POOL BIAS
                                                PBLOX   - NETWORK POOL SIZE IN 32. WORD BLOCKS
                                        ;
                                        ; OUTPUTS: (NO RETURN ON ERROR)
                                                UMRPTR  - UMR BLOCK ADDRESS OR 0
                                                UMR BLOCK WRITTEN TO SYSTEM IMAGE
                                                PUMR INITIALIZED  (NON MULTIPROCESSOR ONLY)                     ;[MP01]
                                        ;-
002262                                  $UMRAL:
002262  005067  000000G                         CLR     UMRPTR          ; ASSUME FAILURE
002266  005767  000000G                         TST     .MGEXT          ; 11/70 TARGET SYSTEM?
002272  100513                                  BMI     25$             ; IF MI, NO
002274  005767  000000G                         TST     UMRFLG          ; HAVE UMR'S BEEN DISABLED?
002300  001110                                  BNE     25$             ; IF NE, YES
002302  026767  000000G 000000G                 CMP     PBLOX,NONMAP    ; MAKE SURE NON-UMR-MAPPED AREA
002310  101515                                  BLOS    121$            ; ISN'T BIGGER THAN POOL SIZE
002312  012701  000014                          MOV     #M.LGTH,R1      ; GET UMR BLOCK SIZE
002316                                          CALL    $ALL15          ; TRY TO ALLOCATE FROM POOL
002322  103500                                  BCS     101$            ; IF CS, FAILURE
002324  010067  000000G                         MOV     R0,UMRPTR       ; SAVE UMR BLOCK ADDRESS
002330  012701  000006'                         MOV     #UMRBF,R1       ; POINT AT LOCAL BUFFER
002334  016702  000000G                         MOV     PBLOX,R2        ; GET POOL SIZE IN BLOCKS
002340  166702  000000G                         SUB     NONMAP,R2       ; LESS NTL BLOCKS, ETC.
002344  166702  000000G                         SUB     .ALLOC,R2       ;    AND EXTENDED POOL ALLOCATIONS
002350  062702  000177                          ADD     #177,R2         ; ROUND UP TO NEXT MULTIPLE OF 4K WORDS
002354  042702  000177                          BIC     #177,R2         ; ...
002360  006202                                  ASR     R2              ; GIVING NUMBER OF UMR'S NEEDED * 4
002362  006202                                  ASR     R2              ; ...
002364  006202                                  ASR     R2              ; ...
002366  006202                                  ASR     R2              ; ...
002370  006202                                  ASR     R2              ; ...
002372  000241                                  CLC                     ; ...
002374  010261  000004                          MOV     R2,M.UMRN(R1)   ; STASH COUNT IN UMR BLOCK
002400  016703  000000G                         MOV     PSTART,R3       ; GET POOL STARTING BLOCK NUMBER
002404  066703  000000G                         ADD     NONMAP,R3       ; LESS NTL BLOCKS, ETC.
002410  005002                                  CLR     R2              ; CONVERT TO A REAL MEMORY ADDRESS
002412  012704  000006                          MOV     #6,R4           ; DOUBLE SHIFT LEFT 6 PLACES
002416  006303                          10$:    ASL     R3              ;       ...
002420  006102                                  ROL     R2              ;       ...
002422  005304                                  DEC     R4              ;
002424  001374                                  BNE     10$             ; ...
002426  010361  000012                          MOV     R3,M.BFVL(R1)   ; STASH IT IN UMR BLOCK
002432  110261  000011                          MOVB    R2,M.BFVH(R1)   ;       ...

                                                .IF DF  R$$MPL                                                  ;[MP01]

                                                TST     .MPRO           ; MULITPROCESSOR ?                      ;[MP01]
                                                BLT     15$             ; IF NO - BRANCH                        ;[MP01]
                                                MOV     $CPBIA,$RBLCK   ; MAP TO LOWER FOR THIS CPU             ;[MP01]
                                                CALL    $ASUM1          ; TRY TO ASSIGN CONTIGUOUS UMR'S        ;[MP01]
                                                BCS     111$            ; IF CS, FAILURE
```

```

003632                          ISTAT$  IBUF3,KBUF3

003632                          STATE$
003632                           TRAN$  %LLC$DF%

003632                          STATE$
003632                           TRAN$  %XPT%

003632                          STATE$
003632                           TRAN$  <',>

003632                          STATE$  XPTDF1
003632                           TRAN$  '!,XPTDF1
003632                           TRAN$  $RAD50,XPTDF1
003632                           TRAN$  <',>

003632                          STATE$
003632                           TRAN$  $NUMBR

003632                          STATE$
003632                           TRAN$  <',>

003632                          STATE$
003632                           TRAN$  $NUMBR,$EXIT,XPCIR

003632                          STATE$

```

```
                                        .SBTTL  OPEN CONFIGURATION FILE

                                ;+
                                ;       OPNCFG
                                ;
                                ; INPUTS:
                                ;       PROPERLY INITIALIZED FDB
                                ; OUTPUTS:
                                ;       C-BIT SUCCESS/FAILURE
                                ;
                                ;-

005560  012700  000000G         OPNCFG: MOV     #$DFUIC,R0      ; SETUP ASCII FILENAME
005564  012701  000000G                 MOV     #$CET,R1        ; USE 'CETAB' NAME
005570  012702  000000G                 MOV     #$MAC,R2        ; USE .MAC EXTENSION
005574                                  CALL    $AZFS           ; ...
005600                                  CALL    $CLQUE          ; ATTEMPT TO OPEN FILE
005604                                  RETURN
```

```



                                      ;+
                                      ; DEC$DF MACRO STATE TABLE
                                      ;-

006324                                        STATE$  DECDF
006324                                         TRAN$  $NUMBR,,ECLLNK          ; MAX LOGICAL LINKS

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,NODCT           ; MAX NODE COUNTERS

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,INCT            ; INCOMING TIMER

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,OUTT            ; OUTGOING TIMER

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,INAC            ; INACTIVITY TIMER

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,DELF            ; DELAY FACTOR

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,DELW            ; DELAT WEIGHT

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,IPL             ; INPUT PACKET LIMITER

006324                                        STATE$
006324                                         TRAN$  <',>

006324                                        STATE$
006324                                         TRAN$  $NUMBR,,LST             ; LINK SERVICE THRESHOLD
```

```
  2528                                              .SBTTL  TPARS ACTION ROUTINES
  2529
  2530                                              .SBTTL  ACTION ROUTINE: BUF$DF MACRO SEEN
  2531                                      ;+
  2532                                      ;       **-.BUFDF-BUF$DF MACRO SEEN
  2533                                      ;
  2534                                      ;       THE BUF$DF MACRO IS USED TO CONTROL THE SIZE AND NUMBER OF
  2535                                      ;       RESOURCES ALLOCATED TO THE NETWORK. HERE WE SET UP THE TABLE
  2536                                      ;       POINTERS TO BE USED WHEN RETRIEVING THE ARGUMENTS FROM THE
  2537                                      ;       MACRO.
  2538                                      ;-
  2539                                      ;
  2540 006324  012767  006404' 000000G .BUFDF: MOV     #DFTBL,DFPTR    ; SET UP THE DEFINITION TABLE POINTER
  2541 006332  012767  000010  000000G         MOV     #DFSIZ,DFCNT    ; MAX # OF ARGUMENTS ALLOWED
  2542 006340                                  RETURN
  2543                                      ;
```

```
                                          .SBTTL  ACTION ROUTINE: HOST NODE ADDRESS SEEN
                                   ;+
                                   ; *** - .HSTAD - HOST NODE ADDRESS SEEN
                                   ;
                                   ; INPUT:
                                   ;       .PNUMB = THE NODE ADDRESS
                                   ;
                                   ;-

007416  022767  001777  000000G .HSTAD: CMP     #1023.,.PNUMB           ; CHECK LIMIT
007424  002416                          BLT     10$                     ; IF ERROR - BRANCH

007426                                  GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
007446  056767  000000G 000022G         BIS     .PNUMB,$BFR+D$HOST      ; STORE THE NODE NUMBER

007454                                  PUTRC                           ; WRITE BACK DECNET HOME BLOCK

007460                                  RETURN

007462  062716  000002          10$:    ADD     #2,(SP)                 ; ILLEGAL HOST NODE ADDRESS
007466                                  RETURN

                                   ;+
                                   ; *** - .HSTAR - HOST AREA
                                   ;-
007470  022767  000077  000000G .HSTAR: CMP     #63.,.PNUMB             ; CHECK LIMIT
007476  002433                          BLT     10$                     ; IF ERROR - BRANCH

007500  016700  000000G                 MOV     .PNUMB,R0               ; COPY AREA
        000012                          .REPT   10.
                                        ASL     R0                      ; POSITION AREA BITS
                                        .ENDR
007530  000241                          CLC

007532                                  GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK

007552  010067  000022G                 MOV     R0,$BFR+D$HOST          ; STORE HOST AREA
007556  001403                          BEQ     10$                     ; IF ERROR - BRANCH

007560                                  PUTRC                           ; WRITE BACK DECNET HOME BLOCK

007564                                  RETURN

007566  062716  000002          10$:    ADD     #2,(SP)                 ; ILLEGAL HOST AREA
007572                                  RETURN

```

```
                                                .SBTTL  ACTION ROUTINE: ROUTING TIMER

                                        ;
                                        ; .RRTM - ROUTING TIMER ACTION ROUTINE
                                        ;
012110  005767  000000G                 .RRTM:  TST     .PNUMH          ; LEGAL ROUTING TIMER INTERVAL VALUE?
012114  001016                                  BNE     10$             ; BR IF NO

012116                                          GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK
012136  016767  000000G 000076G                 MOV     .PNUMB,$BFR+D$RTMR ; GET ROUTING TIMER INTERVAL VALUE

012144                                          PUTRC                   ; WRITE DECNET HOME BLOCK TO IMAGE
012150                                          RETURN
                                        ;
012152  062716  000002                  10$:    ADD     #2,(SP)         ; ILLEGAL ROUTING TIMER INTERVAL VALUE
012156                                          RETURN

```

```
                                          .SBTTL  ACTION ROUTINE: OBJECT COPIES SEEN
                                  ;+
                                  ; *** - .OCPY - OBJECT COPIES SEEN
                                  ;
                                  ; INPUTS:
                                  ;       .PNUMB = OBJECT COPIES SEEN
                                  ;
                                  ; OUTPUTS:
                                  ;       TRANSITION IS ACCEPTED/REJECTED
                                  ;
                                  ;-

                                          .ENABL  LSB
013434  116767  000000G 000064'   .OCPY:: MOVB    .PNUMB,MXCP         ; GET NUMBER OF COPIES
013442  005767  000000G                   TST     .PNUMH              ; IS IT A WORD VALUE ?
013446  001062                            BNE     101$                ; IF NE, NO
013450  105767  000001G                   TSTB    .PNUMB+1            ; IS IT A BYTE VALUE ?
013454  001057                            BNE     101$                ; IF NE, NO
                                  ;
013456  012701  000012            .OEND:: MOV     #O.LEN,R1           ; GET  LENGTH OF AN OBJECT BLOCK
013462                                    CALL    $ALPOL              ; TRY TO ALLOCATE AN OBJECT BLOCK
013466  103455                            BCS     111$                ; IF CS, COULDN'T

013470  012701  000000G                   MOV     #$BFR,R1            ; POINT AT I/O BUFFER
013474  116761  000063' 000002            MOVB    TYPE,O.TYP(R1)      ; STORE THE OBJECT TYPE NUMBER
013502  016761  000032' 000006            MOV     ONAME,O.NAM(R1)     ; STORE THE OBJECT NAME
013510  016761  000034' 000010            MOV     ONAME+2,O.NAM+2(R1) ; ...
013516  116761  000062' 000003            MOVB    FLAG,O.FLG(R1)      ; STORE THE FLAGS BYTE
013524  116761  000064' 000004            MOVB    MXCP,O.MXC(R1)      ; STORE THE MAX NUMBER OF COPIES TO SPAWN
013532  010067  000004'                   MOV     R0,CURUNM           ; SAVE UNMAPPED ADDRESS
013536                                    CEACC$  R0                  ; CONVERT TO MAPPED ADDRESS
013546                            10$:    PUTAD   R0,#O.LEN           ; WRITE OUT THE OBJECT BLOCK
                                  ;
                                  ; INSERT ENTRY INTO LIST
                                  ;
013564  005001                            CLR     R1                  ; GET OBJECT TYPE NUMBER
013566  156701  000002G                   BISB    $BFR+O.TYP,R1       ; ..WITHOUT SIGN EXTEND
013572  016700  000004'                   MOV     CURUNM,R0           ; RETRIEVE UNMAPPED ADDRESS
013576                                    CALL    FINOBJ              ; FIND WHERE OBJECT GOES IN LIST
013602  012702  000000G                   MOV     #$OBJHD,R2          ; POINT TO LISTHEAD
013606                                    CALL    QINS                ; INSERT ENTRY INTO LIST
013612                                    RETURN                      ; RETURN TO USER MODE AND TO CALLER
                                  ;
                                  ; ERROR CONDITIONS
                                  ;
013614  062716  000002            101$:   ADD     #2,(SP)             ; REJECT TRANSITION
013620                                    RETURN
013622                            111$:   NTIERR  23                  ; OBJECT BLOCK ALLOCATION FAILURE
                                          .DSABL  LSB
```

```

015436  005741                    40$:    TST     -(R1)           ; CALCULATE PDV INDEX
015440  166701  000000G                   SUB     $PDVTA,R1       ; ...
015444  005726                            TST     (SP)+           ; CLEAN UP STACK

015446                            50$:    RETURN

```

```
  4662 017216  001004                          BNE     1$              ; IF YES - BRANCH
  4663
  4664 017220  112722  000054                  MOVB    #',,(R2)+       ; STORE GROUP/USER UIC SEPARATOR
  4665 017224  105267  000206'                 INCB    UICBUF          ; INCLUDE ,
  4666
  4667 017230  005067  000000'         1$:     CLR     NAMED           ; INIT NAMED DIRECTORY FLAG
  4668 017234  112022                          MOVB    (R0)+,(R2)+     ; STORE STRING
  4669 017236  105267  000206'                 INCB    UICBUF          ; COUNT ONE CHARACTER
  4670 017242  005301                          DEC     R1              ; ...
  4671 017244  001371                          BNE     1$              ; IF MORE - BRANCH
  4672
  4673 017246  112712  000135                  MOVB    #'],(R2)        ; STORE UIC TERMINATOR CHARACTER
  4674 017252  105267  000206'                 INCB    UICBUF          ; ADJUST UIC STRING SIZE
  4675
  4676 017256  012767  000266' 000334' 10$:    MOV     #FNBUFF,STRPT   ; RESET PARSED STRING POINTER
  4677 017264  005067  000204'                 CLR     STRSIZ          ; INIT SIZE
  4678
  4679 017270                                  RETURN
  4680
  4681
  4682
  4683
```

```
                                        .SBTTL  SPFILL - Fill area with spaces                                  ;[TD001]
                                ;+                                                                              ;[TD001]
                                ; *** SPFILL - Fill area with spaces                                            ;[TD001]
                                ;                                                                               ;[TD001]
                                ; INPUTS                                                                        ;[TD001]
                                ;       R0 - address of area to fill                                            ;[TD001]
                                ;       R1 - length of area to fill                                             ;[TD001]
                                ;                                                                               ;[TD001]
                                ; OUTPUTS                                                                       ;[TD001]
                                ;       All registers preserved                                                 ;[TD001]
                                ;-                                                                              ;[TD001]
006536                          SPFILL:                                                                         ;[TD001]
006536  010046                          MOV     R0,-(SP)                ; save registers                        ;[TD001]
006540  010146                          MOV     R1,-(SP)                                                        ;[TD001]
006542  001404                          BEQ     20$                     ; nothing to do?                        ;[TD001]
006544  112720  000040          10$:    MOVB    #SPACE,(R0)+                                                    ;[TD001]
006550  005301                          DEC     R1                                                              ;[TD001]
006552  003374                          BGT     10$                                                             ;[TD001]
006554  012601                  20$:    MOV     (SP)+,R1                ; restore regs                          ;[TD001]
006556  012600                          MOV     (SP)+,R0                                                        ;[TD001]
006560                                  RETURN                                                                  ;[TD001]
        000001                          .END
```

```
SYMBOL CROSS REFERENCE                                    CREF   04.00

SYMBOL  VALUE              REFERENCES

$IFX9S     002114 RG       #7-608
$INA       002516 RG       #7-705
$INC       002402 RG       #7-683
$LNAME     002166 RG       #7-635
$LTM       002230 RG       #7-650
$MACT      002266 RG       #7-660
$MDEA      002334 RG       #7-672
$MUL     = ******   GX      7-909
$MXCO      002632 RG       #7-727
$MXLI      002700 RG       #7-739
$NADD      003062 RG       #7-777
$NNAM      004722 RG       #7-1088
$ONAM      003214 RG       #7-802
$OPTON   = ******   GX     *6-200      7-384      7-596      7-600      7-604      7-608      7-1277
$OTYP      003260 RG       #7-814
$OUT       002450 RG       #7-694
$OVER      003350 RG       #7-830
$OWN       003360 RG       #7-837
$PCIR      003420 RG       #7-847
$PDT       001434 RG       #7-495
$PDVTA   = ******   GX      11-1402
$PFLAG   = ******   GX      13-1476
$PNAM      003476 RG       #7-864
$PRI       003536 RG       #7-874
$QUAL    = ******   GX     *6-202      7-443      7-1191     7-1227
$RADDR   = ******   GX     *7-549     *7-781     *7-953     *7-1004    *7-1065    *8-1314
$RDBSZ   = ******   GX      7-931
$RET       002564 RG       #7-716
$SAVVR   = ******   GX      7-1202     7-1238     7-1270
$SCOND     003636 RG       #7-895
$SCOP1     003606 RG       #7-886
$SCOP2     003616 RG       #7-889
$SCOP3     003626 RG       #7-892
$SCSR      003750 RG       #7-918
$SEG       004026 RG       #7-931
$SLTMA   = ******   GX      11-1395
$SLTNM   = ******   GX      11-1396
$STADD     004102 RG       #7-948
$STALI     004234 RG       #7-974
$STDES     004246 RG       #7-982
$STEND     004260 RG       #7-990
$STHOS     004306 RG       #7-1000
$STLND     004440 RG       #7-1026
$STLSH     004552 RG       #7-1054
$STLSN     004516 RG       #7-1046
$STNAD     004570 RG       #7-1061
$STNAM     004734 RG       #7-1095
$STPAR     005026 RG       #7-1121
$STRPA     005072 RG       #7-1132
$STSNM     005166 RG       #7-1153
$STTPA     005230 RG       #7-1169
$TRI       005324 RG       #7-1191
```

```
    249 000402  111667  000000G                  MOVB    (SP),$OPTON               ; RESTORE LOW BYTE OF OPTIONS WORD
    250 000406  000720                           BR      15$                       ; ELSE, GET THE NEXT LINE
    251
    252 000410  005726                   60$:    TST     (SP)+                     ; CLEAN UP STACK
    253 000412  032767  000004  000000G          BIT     #QF$KNO,$QUAL             ; WAS THIS FOR KNOWN LINES ?
    254 000420  001002                           BNE     70$                       ; IF NE, YES .. LEAVE
    255 000422  005704                           TST     R4                        ; ELSE, WERE ANY LINES FOUND
    256 000424  001405                           BEQ     102$                      ; IF EQ, NO .. LINE NOT IN SYSTEM
    257 000426                           70$:    RETURN
    258                                  ;
    259                                  ; ERROR CONDITIONS
    260                                  ;
    261 000430                           101$:   ERRPT$  LNER1,$EROUT              ; INVALID LINE OPTION
    262 000440                           102$:   ERRPT$  LNER2,$EROUT              ; LINE NOT IN SYSTEM
```

```
                                        .SBTTL  $SLIN - SET LINE
                                ;+
                                ; $SLIN - "SET LINE"
                                ;
                                ; INPUTS:
                                ;       $OPTON = OPTIONS WORD
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;
                                ;-

002236                          $SLIN:
002236  105767  000000G                 TSTB    $OPTON                  ; DO WE NEED TO LOAD THE LINE ?
002242  001403                          BEQ     10$                     ; IF EQ, NO
002244                                  CALL    LDLIN                   ; ELSE, LOAD THE LINE
002250  103417                          BCS     40$                     ; IF CS, LOAD FAILED
002252  012705  002312'         10$:    MOV     #LINTAB,R5              ; GET FUNCTION DISPATCH TABLE
002256  012504                  20$:    MOV     (R5)+,R4                ; GET OPTIONS BIT TO CHECK
002260  001413                          BEQ     40$                     ; BR IF END OF TABLE
002262  030467  000000G                 BIT     R4,$OPTON               ; WAS THIS OPTION SPECIFIED?
002266  001406                          BEQ     30$                     ; BR IF NO
002270  010446                          MOV     R4,-(SP)                ; SAVE R4
002272  010546                          MOV     R5,-(SP)                ; SAVE R5
002274                                  CALL    @(R5)                   ; ELSE, EXECUTE REQUEST
002300  012605                          MOV     (SP)+,R5                ; RESTORE R5
002302  012604                          MOV     (SP)+,R4                ; RESTORE R4
002304  005725                  30$:    TST     (R5)+                   ; SKIP TO NEXT REQUEST
002306  000763                          BR      20$                     ; IF NE, YES
002310                          40$:    RETURN

002312                          LINTAB:
002312  000400  000000G                 .WORD   OP$MDE,$VDEA            ; SET MULTIPOINT DEAD RATIO
002316  000000                          .WORD   0
```

```
                                                                  E  5
VNP         CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 8
MACRO CROSS REFERENCE                                      CREF    04.00

MACRO NAME       REFERENCES

ASR$             #5-60      15-526
CALL              7-143     7-146      7-148      10-217     10-247     10-261     10-262     11-277     11-278     11-280
                  11-286    11-290     11-295     11-306     12-335     12-373     12-390     15-511     15-521     15-533
                  16-573    16-581     16-585     16-592     16-594     16-596     16-598     16-601     16-607     17-644
                  17-651    17-654     17-656     17-661     17-666     18-696     18-705     19-736     19-739     19-741
                  19-744    19-747     19-750     19-752     19-760     19-763     19-767     20-790
CALLR             16-620    19-774
ERBLK$           #5-60      6-84       6-86       6-88       11-310     15-542     15-544
ERMSG$           #5-60      6-83       6-85       6-87       11-309     15-541     15-543
ERRPT$           #5-60      10-261     10-262     11-306     12-390     15-521     15-533
GETRV            #5-60      20-790
MOVIT            #15-502    15-516     15-520     15-528     15-532
RAND             #20-790    20-790
RETURN            10-257    11-302     11-307     12-386     14-492     15-535     15-537     16-614     17-668     18-710
                  19-769    20-792
VNPDF$           #5-60      5-69
```

```
                                                                  F  5
**FILE**ID**VNSD
```

```
                                      .SBTTL  LOCAL DATA


                               ;
                               ; LOCAL DATA
                               ;
 000000                               .PSECT  DATA,D
 000000  000000                NAMED:  .WORD   0               ; NAMED DIRECTORY FLAG
 000002                        PRVUNM: .BLKW   1               ; UNMAPPED ADDRESS OF PREVIOUS BLOCK IN LIST
 000004                        CURUNM: .BLKW   1               ; UNMAPPED ADDRESS OF CURRENT BLOCK IN LIST
 000006                        UMRBF:  .BLKW   M.LGTH/2        ; LOCAL COPY OF UMR BLOCK
 000022                        NAMBUF: .BLKW   3               ; ASCII TO RAD50 CONVERSION BUFFER
 000030                        NAME:   .BLKW   1               ; REMOTE NODE NAME
 000032                        ONAME:  .BLKW   2               ; OBJECT TASK NAME
 000036                        STAN:   .BLKW   1               ; STATION NUMBER
 000040                        MRKFL:  .BLKW   1               ; MARK FOR LOAD FLAG
 000042                        CLASS:  .BLKW   1               ; EVENT CLASS NUMBER
 000044                        FLTFLG: .BLKW   1               ; FILTER BLOCK FLAGS WORD
 000046                        EVENT:  .BLKW   4               ; EVENT TYPE MASKS
 000056                        STRADD: .BLKW   1               ; ADDRESS OF CURRENT EVENT MASK TO PROCESS
 000060                        CTL:    .BLKB   1               ; CONTROLLER NUMBER
 000061                        UNT:    .BLKB   1               ; UNIT NUMBER
 000062                        FLAG:   .BLKB   1               ; OBJECT FLAGS BYTE
 000063                        TYPE:   .BLKB   1               ; OBJECT TYPE
 000064                        MXCP:   .BLKB   1               ; OBJECT MAX. NUMBER OF COPIES
                                       .EVEN

 000066  000000                XPTLLN: .WORD   0               ; XPT CIRCUITS
 000070  000000                ECLLLN: .WORD   0               ; MAX LOGICAL LINKS
 000072  000000                ECLNOD: .WORD   0               ; ECL NODE COUNTERS

 000074                        NN:     .BLKW   1               ; MAX NOD ADDRESS IN SYSTEM
 000076                        NA:     .BLKW   1               ; MAX AREAS
 000100                        REMARE: .BLKW   1               ; REMOTE AREA
 000102  000000                NBRA:   .WORD   0               ; TOTAL OF BROADCAST ROUTER ADJACENCIES
 000104  000000                RAD:    .WORD   0               ; INTERMEDISTE ROUTER ADJACENCY VALUE
 000106  000000                NBEAJ:  .WORD   0               ; BROADCAST ENDNODE ADJACENCY VANLUE

```

```
    727                                               BR      20$            ; IF SUCCESS - BRANCH                                 ;[MP01]
    728                                               .ENDC                                                                         ;[MP01]
    729
    730 002436                              15$:      CALL    $ASUMR         ; ASSIGN UMR'S                                        ;[MP01]
    731
    732 002442  016102  000002                        MOV     M.UMRA(R1),R2  ; CALCULATE AND SAVE EVENTUAL $PUMR VALUE
    733 002446  162702  170200                        SUB     #UBMPR,R2      ;        ...
    734 002452  006302                                ASL     R2             ;        ...
    735 002454  006302                                ASL     R2             ;        ...
    736 002456  006302                                ASL     R2             ;        ...
    737 002460  006302                                ASL     R2             ;        ...
    738 002462  006302                                ASL     R2             ;        ...
    739 002464  000241                                CLC                    ;        ...
    740 002466  010267  000000G                       MOV     R2,PUMR        ;        ...
    741 002472                              20$:      SWTCHO  R1             ; CHANGE DEFAULT BUFFER
    742 002476                                        PUTRC   R0,#M.LGTH     ; WRITE UMR BLOCK
    743 002514                                        SWTCHO                 ; RESET DEFAULT BUFFER
    744
    745 002522                              25$:      RETURN
    746
    747                                     ;
    748                                     ; ERROR CONDITIONS
    749                                     ;
    750 002524                              101$:     NTIERR  20             ; UMR BLOCK ALLOCATION FAILURE
    751 002534                              111$:     NTIERR  21             ; NOT ENOUGH CONTIGUOUS UMR'S
    752 002544                              121$:     NTIERR  22             ; NON-UMR-MAPPED AREA TOO LARGE
```

```
   1248
   1249
   1250 003632                            ISTAT$  IBUF4,KBUF4
   1251
   1252 003632                            STATE$
   1253 003632                              TRAN$  %DEC$DF%
   1254 003632                            STATE$
   1255 003632                              TRAN$  $NUMBR,,ECLLNK
   1256 003632                            STATE$
   1257 003632                              TRAN$  <',>
   1258 003632                            STATE$
   1259 003632                              TRAN$  $NUMBR,$EXIT,NODC
   1260
   1261 003632                            STATE$
   1262
```

```
                                                .SBTTL  SCAN CONFIG FILE FOR NTPOOL DEFINITIONS
                                        ;+
                                        ;       NTPSCN
                                        ;
                                        ; INPUTS:
                                        ;       THE CONFIG FILE SHOULD BE OPEN
                                        ;
                                        ; OUTPUTS:
                                        ;       NTPNAM  - RETURNED AS THE RAD50 NAME FOUND IN THE BUF$DF MACRO
                                        ; IF MAPPED
                                        ;       NTPSIZ  - NUMBER OF 32-WORD BLOCKS TO ALLOCATE
                                        ;       NTPPAR  - NTPOOL SYSTEM PARTITION NAME TO USE WHEN ALLOCATING
                                        ;-

005606                                  NTPSCN:
005606  005067  000000G                         CLR     NTPSIZ          ; ZERO POOL SIZE ACCUMULATOR
005612  005067  000002G                         CLR     NTPSIZ+2        ; BOTH WORDS
005616  012767  000001  000000G                 MOV     #1,NONMAP       ; DEFAULT SIZE OF NON-UMR-MAPPED AREA
005624  012767  160000  000000G                 MOV     #160000,CFSFLG  ; INITIALIZE FLAG WORD
005632                                  10$:    GCL$                    ; READ IN NEXT RECORD
005636  103546                                  BCS     60$             ; READ ERROR
005640  016703  000000G                         MOV     $CLIOS,R3       ; IOSB ADDRESS
005644  016303  000002                          MOV     2(R3),R3        ; NUMBER OF CHARACTERS READ
005650  010367  000000G                         MOV     R3,LINSIZ       ; STORE LINE SIZE IN CASE OF ERROR
005654  001766                                  BEQ     10$             ; SKIP ZERO LENGTH LINES


                                        ;+
                                        ; LLC$DF (ECL,XPT)
                                        ;
                                        ; LLC$DF ARE PARSED FOR VALUES IMPACTING THE SIZE OF POOL.
                                        ;
                                        ;-

005656  012702  000014'                         MOV     #KBUF3,R2       ; SET UP KEYWORD ADDRESS XPT STATE TABLE
005662  012705  000254'                         MOV     #IBUF3,R5
005666                                          CALL    PARSE           ; LOOK FOR XPT
005672  012702  000020'                         MOV     #KBUF4,R2       ; SET UP KEYWORD ADDRESS ECL STATE TABLE
005676  012705  000306'                         MOV     #IBUF4,R5       ; LOOK FOR ECL
005702                                          CALL    PARSE
005706  012702  000022'                         MOV     #KBUF5,R2       ; LOOK FOR SLT$DF
005712  012705  000324'                         MOV     #IBUF5,R5       ; SET UP START OF STATE TABLE
005716                                          CALL    PARSE

                                        ;
                                        ; ROU$DF
                                        ;
005722  032767  020000  000000G                 BIT     #20000,CFSFLG   ; HAS ROU$DF ALREADY BEEN PROCESSED?
005730  001424                                  BEQ     18$             ; IF EQ, YES
005732  012702  000000'                         MOV     #KBUF0,R2       ; LOOK FOR ROU$DF
005736  012705  000000'                         MOV     #IBUF0,R5       ; ...
005742                                          CALL    PARSE           ; ...
005746  103404                                  BCS     15$             ; IF CS, ERROR
005750  042767  020001  000000G                 BIC     #20001,CFSFLG   ; INDICATE ROU$DF SEEN WITHOUT ERROR
005756  001473                                  BEQ     50$             ; IF EQ, ALL DONE
005760  105767  000000G                 15$:    TSTB    CFSFLG          ; WAS THE ROU$DF IN ERROR ?
005764  001406                                  BEQ     18$             ; IF EQ, NO
```

```
  2162
  2163 006324                                  STATE$
  2164 006324                                   TRAN$  <',>
  2165
  2166 006324                                  STATE$
  2167 006324                                   TRAN$  $NUMBR,,RETF              ; RETRANSMIT FACTOR
  2168
  2169 006324                                  STATE$
  2170 006324                                   TRAN$  <',>
  2171
  2172 006324                                  STATE$
  2173 006324                                   TRAN$  $NUMBR,$EXIT,SEG          ; ECL SEGMENT SIZE
  2174
  2175 006324                                  STATE$
  2176
```

```
                                         .SBTTL  ACTION ROUTINE: BUF$DF ARGUMENT MISSING
                                 ;+
                                 ;       **-.BUFSK-BUF$DF ARGUMENT MISSING
                                 ;
                                 ;       ONE OF THE ARGUMENTS OF THE BUF$DF MACRO WAS FOUND TO BE MISSING,
                                 ;       WE WILL USE THE DEFAULT VALUE PLACED IN THE COMMEXEC DATABASE
                                 ;       AT SYSTEM BUILD TIME AND MOVE ON THE POINTERS FOR THE NEXT
                                 ;       ARGUMENT.
                                 ;-
                                 ;
006342  062767  000002  000000G  .BUFSK: ADD     #2,DFPTR        ; MOVE ON THE DEFINITION POINTER
006350  005367  000000G                  DEC     DFCNT           ; ANOTHER ARGUMENT SCANNED
006354                                   RETURN
                                 ;
```

```
                                                .SBTTL  ACTION ROUTINE: SYSTEM ID SEEN
                                        ;+
                                        ;       **-.SYSID-SYSTEM ID SEEN
                                        ;
                                        ;       THE SYSTEM ID STRING HAS BEEN FOUND IN THE 'NOD$DF' MACRO.
                                        ;-
                                        ;
007574  016767  000000G 000434          .SYSID: MOV     .PSTPT,STRPTR   ; SAVE ADDRESS OF NEXT BYTE
007602  005267  000430                          INC     STRPTR          ;   WE ARE POINTED AT THE "<" NOW
007606  012701  000042                          MOV     #34.,R1         ; ALLOCATE POOL.. FOR ID STRING

007612                                  1$:     CALL    $ALPOL
007616                                  5$:     SWTCHI  #$HMBUF         ; USE TEMPORARY BUFFER
007624                                          GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK
007644                                          SWTCHI                  ; RESET DEFAULT BUFFER

007652                                          SWTCHO  #$HMBUF         ; USE TEMPORARY BUFFER
007660  010067  000020G                         MOV     R0,$HMBUF+D$LID ; SET UP UNMAPPED ADDRESS FOR ID STRING
007664                                          PUTRC                   ; UPDATE HOME BLOCK
007670                                          SWTCHO                  ; RESET DEFAULT BUFFER

007676  012767  000040  000334                  MOV     #32.,STRMAX     ; MAXIMUM TEXT LENGTH
007704                                          RETURN                  ; WAIT FOR THE ">" (.STRNG)
                                        ;
```

```
                                                 .SBTTL  ACTION ROUTINE: MAXIMUM AREA

                                          ;
                                          ; .RMAXA - MAXIMUM AREA ACTION ROUTINE
                                          ;
012160                                    .RMAXA: GETRV   $DECPT,#D$END     ; READ DECNET HOME BLOCK

012200  005767  000060G                           TST     $BFR+D$NN         ; ROUTING NODE ?
012204  001004                                    BNE     10$               ; IF YES - BRANCH
012206  005767  000000G                           TST     .PNUMB            ; IF NO - VALUE MUST BE ZERO
012212  001405                                    BEQ     20$               ; IF LEGAL - BRANCH
012214  000412                                    BR      30$               ; ELSE - REPORT ERROR
012216  022767  000077  000000G 10$:              CMP     #63.,.PNUMB       ; CHECK FOR HIGH LIMIT
012224  103406                                    BLO     30$               ; IF OUT OF RANGE - BRANCH

012226  016767  000000G 000062G 20$:              MOV     .PNUMB,$BFR+D$NA  ; COPY NUMBER OF AREAS IN NETWORK

012234                                            PUTRC                     ; WRITE BACK DECNET HOME BLOCK

012240                                            RETURN

012242  062716  000002          30$:              ADD     #2,(SP)           ; ILLEGAL MAX AREA VALUE
012246                                            RETURN

```

```
   3705                                            .SBTTL  FINOBJ - FIND PLACE TO INSERT OBJECT
   3706                                    ;+
   3707                                    ; *** - FINOBJ - FIND PLACE TO INSERT OBJECT
   3708                                    ;
   3709                                    ; INPUTS:
   3710                                    ;       TYPE = OBJECT TYPE NUMBER TO BE INSERTED
   3711                                    ;
   3712                                    ; OUTPUTS:
   3713                                    ;       R0 = PRESERVED
   3714                                    ;       R1 = UNMAPPED ADDRESS OF PREVIOUS BLOCK
   3715                                    ;       R2 = DESTROYED
   3716                                    ;-
   3717
   3718 013632                             FINOBJ:
   3719 013632  012702  000000G                    MOV     #$OBJHD,R2              ; GET START OF OBJECTS LIST
   3720 013636  011267  000000G                    MOV     (R2),$BFR               ; GET FIRST OBJECT IN LIST
   3721 013642  010267  000004'                    MOV     R2,CURUNM               ; INITIALIZE CURRENT UNMAPPED ADDRESS
   3722 013646  016767  000004' 000002'    10$:    MOV     CURUNM,PRVUNM           ; SAVE UNMAPPED ADDRESS OF PREVIOUS
   3723 013654  016767  000000G 000004'            MOV     $BFR,CURUNM             ; GET UNMAPPED ADDRESS OF NEXT IN LIST
   3724 013662  001420                             BEQ     30$                     ; IF EQ, NO MORE
   3725 013664                                     CEACC$  CURUNM,R2               ; CONVERT TO MAPPED ADDRESS
   3726 013676                             20$:    GETAD   R2,#O.LEN               ; READ OBJECT IN INTERNAL BUFFER
   3727 013714  126767  000063' 000002G            CMPB    TYPE,$BFR+O.TYP         ; DOES OBJECT GO HERE ?
   3728 013722  101351                             BHI     10$                     ; IF HI, NO
   3729 013724  016701  000002'            30$:    MOV     PRVUNM,R1               ; RETRIEVE UNMAPPED ADDRESS OF PREVIOUS
   3730 013730                                     RETURN
```

```

                                        ;+
                                        ; FSLT - FIND SLT ADDRESS
                                        ;
                                        ; INPUTS -
                                        ;          R0 - DDM PDV INDEX
                                        ;          R1 - CONTROLLER NUMBER
                                        ;          R2 - UNIT NUMBER
                                        ;
                                        ; OUTPUTS -
                                        ;          SERSLT - CONTAINS SLN
                                        ;
                                        ;-

015450                                  FSLT:   SAVRG   <R3,R4>         ; SAVE REGS
015454  005046                                  CLR     -(SP)           ; CLEAR A WORD ON STACK
015456  110116                                  MOVB    R1,(SP)         ; SET CONTROLLER NUMBER
015460  110266  000001                          MOVB    R2,1(SP)        ; SET UNIT NUMBER
015464  016704  000000G                         MOV     $SLTMA,R4       ; GET SLT VECTOR
015470  012703  177777                          MOV     #-1,R3          ; INIT SLN COUNTER

015474  012402                          10$:    MOV     (R4)+,R2        ; GET NEXT SLT ADDRESS
015476  005203                                  INC     R3              ; COUNT ONE MORE LINE
015500  126703  000000G                         CMPB    $SLTNM,R3       ; END OF SLT'S ?
015504  001002                                  BNE     15$             ; IF YES - ERROR
015506  000261                                  SEC                     ; SET C-BIT
015510  000414                                  BR      30$             ; CONTINUE

015512  126200  000002                  15$:    CMPB    L.DDM(R2),R0    ; MATCH ?
015516  001366                                  BNE     10$             ; IF NO - BRANCH
015520  126216  000012                          CMPB    L.CTL(R2),(SP)  ; CORRECT CONTROLLER ?
015524  001363                                  BNE     10$             ; IF NOT - BRANCH
015526  126266  000013  000001                  CMPB    L.UNT(R2),1(SP) ; CORRECT UNIT ?
015534  001357                                  BNE     10$             ; IF NOT - BRANCH

015536  010367  000262'                         MOV     R3,SERSLT       ; STORE SLN

015542  005726                          30$:    TST     (SP)+           ; CLEAN STACK

015544                                          RESRG   <R4,R3>         ; RESTORE REGS

015550                                          RETURN


```

```
                                        .SBTTL  QINS - INSERT ENTRY INTO QUEUE
                                ;+
                                ; *** - QINS - INSERT ENTRY INTO MIDDLE OF QUEUE
                                ;
                                ; INPUTS:
                                ;       R0 = UNMAPPED ADDRESS OF CURRENT BLOCK TO BE INSERTED
                                ;       R1 = UNMAPPED ADDRESS OF PREVIOUS BLOCK
                                ;       R2 = ADDRESS OF LISTHEAD (INTERNAL STORAGE)
                                ;
                                ; OUTPUTS:
                                ;       R0-R1 = PRESERVED
                                ;       R2 = DESTROYED
                                ;
                                ;-

017272                          QINS:
017272                                  SAVRG   <R0>                    ; SAVE R0
017274  020102                          CMP     R1,R2                   ; IS PREVIOUS ENTRY THE LISTHEAD ?
017276  001022                          BNE     10$                     ; IF NE, NO
017300  011246                          MOV     (R2),-(SP)              ; AND SAVE POINTER TO NEXT ENTRY
017302  010012                          MOV     R0,(R2)                 ; INSERT NEW ENTRY INTO LIST
017304                                  CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
017314                                  GETAD   R0,#2                   ; READ IN POINTER
017332  012667  000000G                 MOV     (SP)+,$BFR              ; STORE POINTER TO NEXT
017336                                  PUTAD                           ; WRITE BACK THE LINK POINTER
017342  000404                          BR      20$                     ; AND EXIT
017344  010046                  10$:    MOV     R0,-(SP)                ; GET UNMAPPED ADDRESS OF CURRENT BLOCK
017346  010146                          MOV     R1,-(SP)                ; GET UNMAPPED ADDRESS OF PREVIOUS
017350                                  CALL    $XLINK                  ; LINK CURRENT BLOCK INTO LIST
017354                          20$:    RESRG   <R0>                    ; RESTORE R0
017356                                  RETURN
```

```
                                                                 F 1
VNCP - VNP GET COMMANDS NCP STY MACRO V05.03b Saturday 29-Jun-   02:44  Page 15-1
Symbol table

ACCTTL   001512R   002 D$LNAM   000006        IFMODT   002162R       LILLSE=  000037        LO.TPA   000020
ADDRS    000102R   002 D$LNUM   000014        ILLARE   002312R   002 LILLTR=  000040        LO.TRB   000034
ARAMSK=  ****** GX     D$LST    000047        ILLCOS   000607R   002 LILLUC=  000037        LO.TRI   000030
AREAN    000100R   002 D$MAXC   000064        ILLCSR   000455R   002 LILLVE=  000037        LO.UNT   000033
ASCII    000064R   002 D$MAXH   000066        ILLINA   002514R   002 LIVDTE=  000032        LO.VCT   000014
A$$CHK=  000000        D$MAXV   000070        ILLINC   002403R   002 LIVLII=  000042        LO.VER   000042
A$$CPS=  000000        D$MLL    000040        ILLMAC   001037R   002 LLNAMT=  000031        LPASWT=  000030
A$$PRI=  000000        D$MNOD   000041        ILLMDE   001105R   002 LMARKE=  000002        LPNAMT=  000034
A$$TRP=  000000        D$NA     000062        ILLNAD   000137R   002 LMSTRI=  000052        LPRNMT=  000036
BDGRP    002047R   002 D$NBEA   000056        ILLOBJ   001253R   002 LMSUNU=  000045        LPSWFT=  000036
BDHELT   002110R   002 D$NBRA   000054        ILLOUT   002561R   002 LMXCN6=  000050        LR.ACC   000020
BDLIST   002144R   002 D$NEND=  000054        ILLPRI   000507R   002 LMXLN6=  000042        LR.ACO   000052
CHKLOG   006366R       D$NLN    000030        ILLRET   002446R   002 LNAMET=  000024        LR.ADD   000012
CKEVE    000452R       D$NN     000060        ILLSEG   002344R   002 LNAMTL   000226R   002 LR.ALI   000002
CKEVEB   000536RG      D$OUTT   000043        ILLTRI   001213R   002 LNSLIN=  000031        LR.BLK   000010
CKEVEC   000566RG      D$RETF   000050        ILLUCS   001000R   002 LNSTER=  000035        LR.CTL   000004
CKEVEN   000640RG      D$RNN    000002        ILLVEC   000550R   002 LOCOPT=  000042        LR.DES   000010
CKEVER   000666RG      D$RTMR   000076        ID.WLB=  ****** GX     LONAMT=  000042        LR.HLP   000002
CM$SHO=  ****** GX     D$SEG    000036        IVDTEA   002225R   002 LOTSKT=  000040        LR.LIN   000002
CNUMTL   000357R   002 D$SER    000032        IVLIID   001407R   002 LO.ADD   000022        LR.NAM   000012
CRLF     000106R   002 D$SQRL   000052        I$$RAR=  000000        LO.ADR   000052        LR.PAS   000041
C$$CKP=  000000        D$$BUG=  177514        I$$RDN=  000000        LO.CLS   000002        LR.PRO   000002
C$$ORE=  000400        D$$ISK=  000000        KEYTAB=  ****** GX     LO.CDN   000006        LR.STS   000000
C$$RSH=  177564        D$$L11=  000001        K$$CNT=  177546        LO.CDP   000010        LR.TRI   000006
DTEATL   002257R   002 D$$YNC=  000000        K$$CSR=  177546        LO.CDS   000020        LR.TYP   000002
DV.CCL=  000002        D$$YNM=  000000        K$$LDC=  000000        LO.CSR   000010        LR.UCB   000016
DV.COM=  020000        D.NAM =  ****** GX     K$$TPS=  000074        LO.CTL   000032        LR.UIC   000020
DV.DIR=  000010        D.PCB =  ****** GX     LACCTT=  000035        LO.DES   000002        LR.UNT   000005
DV.EXT=  000400        D.UCB =  ****** GX     LBDGRP=  000041        LO.DTE   000047        LSYNER=  000023
DV.F11=  040000        D.UCBL=  ****** GX     LBDHEL=  000034        LO.EVT   000004        LS.ALI=  000016
DV.ISP=  002000        D.UNIT=  ****** GX     LBDLIS=  000035        LO.GRD   000024        LS.CEX=  000001
DV.MBC=  000400        EBUF   = 000014R   002 LCNUMT=  000042        LO.HOS   000040        LS.CIR=  000033
DV.MNT=  100000        ELEN   = 000016R   002 LCRLF =  000006        LO.HTM   000032        LS.CXO=  010000
DV.MSD=  000100        EMSG     000000R   002 LDTEAT=  000033        LO.INA   000062        LS.CX5=  000400
DV.OSP=  004000        EOJ      000320R       LD$LP =  000000        LO.INC   000060        LS.ECH=  001000
DV.PSE=  010000        ERRHDL   006304R       LD.ALI=  000030        LO.LDT   000046        LS.FDX=  004000
DV.REC=  000001        ERRLID   005654R       LD.CEX=  000024        LO.LID   000030        LS.HDX=  010000
DV.SDI=  000020        EVEBEG=  000062R   002 LD.CIR=  000035        LO.LIN   000004        LS.HLP=  000012
DV.SQD=  000040        E$$XPR=  000000        LD.LIN=  000025        LO.LTM   000034        LS.LIN=  000003
DV.SWL=  001000        FL.CNT=  ****** GX     LD.LOG=  000031        LO.MAC   000022        LS.LMC=  000007
DV.TTY=  000004        FNDDCB   006146R       LD.MOD=  000037        LO.MDE   000024        LS.LOG=  000020
DV.UMD=  000200        FORWR    004514R       LD.NOD=  000027        LO.NAM   000044        LS.MOD=  000036
D$AMXC   000072        F$$LVL=  000001        LD.OBJ=  000032        LO.NET=  000002        LS.NOD=  000014
D$AMXH   000074        F.RSIZ=  ****** GX     LD.PRO=  000026        LO.NNM   000022        LS.NTI=  000013
D$ANN    000000        F.URBD=  ****** GX     LF.MTP=  ****** GX     LO.OFL   000012        LS.OBJ=  000022
D$BRPR   000102        GBUF   = 000044R   002 LGETFA=  000041        LO.ONM   000004        LS.OPT=  000400
D$BRTM   000100        GETFAL   001451R   002 LILLAR=  000032        LO.OUT   000056        LS.PRO=  000002
D$DELF   000045        GETLIN   006314R       LILLCD=  000030        LO.OWN   000026        LS.TOP=  002000
D$DELW   000046        GLEN   = 000046R   002 LILLCS=  000032        LO.PAR   000010        LS.TST=  000010
D$END =  000104        GMSG     000030R   002 LILLIN=  000045        LO.PRI   000012        LS.UNF=  020000
D$FNB    000034        G$$TPP=  000000        LILLMA=  000046        LO.RET   000064        LS.XIT=  000011
D$HIOR   000024        G$$TSS=  000000        LILLMD=  000044        LO.RPA   000030        LTRIBT=  000041
D$HOST   000022        G$$TTK=  000000        LILLNA=  000033        LO.SCR   000016        LUIDTL=  000035
D$INAC   000044        G$$WRD=  000000        LILLOB=  000032        LO.SEG   000054        LUNUMT=  000034
D$INCT   000042        IE.EDF=  ****** GX     LILLOU=  000043        LO.SKA   000014        LX.ALI=  000017
D$IPL    000051        IFALIT   002152R       LILLPR=  000041        LO.SKN   000014        LX.CEX=  000004
D$LID    000020        IFLDGT   002156R       LILLRE=  000046        LO.SNM   000035        LX.CIR=  000034
```

```
                                                                 G 1
VNCP - VNP GET COMMANDS NCP STY MACRO V05.03b Saturday 29-Jun-85 02:44  Page 15-2
```

```
                                                        F  2
VNCP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44   PAGE 10
SYMBOL CROSS REFERENCE                                CREF   04.00

SYMBOL   VALUE           REFERENCES

$TRIBA    005456 RG      #7-1215
$TYPE   = ******   GX    *6-203
$UNT      005522 RG      #7-1227
$VEC      005674 RG      #7-1254
$VFLAG  = ******   GX    *6-209    *7-552    7-579     7-990     7-1026    *7-1031   7-1046    *7-1051   *7-1055
$VNCP     000000 RG      #6-198
$XMISS    005752 RG      #7-1267
$XTRI     005372 RG       7-1200   #7-1202
$XUNT     005570 RG       7-1236   #7-1238
$$$ARG  = 000003         #5-106     5-106    #5-106     5-106    #5-106     5-106    #5-106     5-106     5-106
                         #5-112     5-112    #5-112     5-112    #5-112     5-112    #5-112     5-112     5-112
$$$GLB  = ******          5-106     5-112
.PCHAR  = ******   GX     7-887     7-890
.PNUMB  = ******   GX     7-321     7-336     7-355     7-368     7-376     7-407     7-408     7-420     7-422
                          7-432     7-434     7-436     7-448     7-452     7-454     7-572     7-653     7-662
                          7-664     7-674     7-676     7-685     7-687     7-696     7-698     7-707     7-709
                          7-718     7-720     7-729     7-741     7-753     7-768     7-817     7-819     7-876
                          7-878     7-893     7-920     7-922     7-924     7-933     7-935     7-938     7-1195
                          7-1199    7-1201    7-1217    7-1219    7-1231    7-1235    7-1237    7-1256    7-1258
                          7-1260
.PNUMH  = ******   GX     7-319     7-334     7-353     7-366     7-405     7-418     7-430     7-446     7-570
                          7-651     7-660     7-672     7-683     7-694     7-705     7-716     7-727     7-739
                          7-815     7-874     7-918     7-1193    7-1215    7-1229    7-1254
.PSTCN  = ******   GX     7-272     7-462     7-479     7-498     7-505     7-513     7-532     7-1101    7-1110
                          7-1134    7-1142    7-1153    7-1171    7-1179    8-1287    13-1479   13-1484   13-1486
                          13-1490
.PSTPT  = ******   GX     7-281     7-465     7-482     7-504     7-535     7-1100    7-1133    7-1160    7-1170
                          8-1286    13-1491
```

```
                                                        G  2
VNCP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44   PAGE 11
```

```
                                                        .SBTTL  $VRQB - PROCESS REQUESTS FROM VNCP
                                              ;+
                                              ; **-$VRQB-PROCESS REQUESTS FROM VNCP
                                              ;
                                              ; INPUTS:
                                              ;         RQB = REQUEST BLOCK FILLED IN AS NECESSARY
                                              ;
                                              ; OUTPUTS:
                                              ;         RQB = STATUS CODE
                                              ;
                                              ;-

       000450                                 $VRQB::
       000450                                         CALL    $SAVAL            ; SAVE ALL REGISTERS
       000454                                         CALL    $CLSAV            ; SAVE GCL CONTEXT
       000460  112767  000000G 000000G                MOVB    #CL.OPN,$CLFLG    ; LEAVE DATA FILE(S) OPEN
       000466                                         CALL    $NLRQB            ; PARSE COMMAND INTO ROOT
       000472  103443                                 BCS     40$               ; COMMAND ERROR
       000474  005767  000000G                        TST     $FUNC             ; IS THIS FUNCTION A NOP ?
       000500  001440                                 BEQ     40$               ; YES
       000502  005767  000000G                        TST     $MISS             ; ARE ANY REQUIRED PARAMETERS MISSING ?
       000506  001403                                 BEQ     20$               ; IF EQ, NO
       000510                                         CALL    $NLCFG            ; SCAN CONFIG FILE
       000514  103416                                 BCS     23$               ;  ERROR...BUT NOT FATAL
       000516  032767  000000G 000000G 20$:           BIT     #VF.ECH,$VFLAG    ; 'SET SYSTEM ECHO' MODE ?
       000524  001402                                 BEQ     22$               ; NO
       000526                                         CALL    ECHO              ; ECHO REQUEST
       000532  016700  000000G         22$:           MOV     $FUNC,R0          ; GET FUNCTION CODE
       000536  006300                                 ASL     R0                ; CONVERT TO WORD INDEX
       000540  016000  001122'                        MOV     FTAB-2(R0),R0     ; GET ADDRESS OF ROUTINE FOR FUNCTION
       000544  001417                                 BEQ     101$              ; IF EQ, NONE .. MUST BE UNSUPPORTED COMMAND
       000546                                         CALL    (R0)              ; TRANSFER TO FUNCTION SPECIFIC ROUTINE
       000550  103006                                 BCC     25$               ; NO PROBLEM
       000552  012767  177777  000000G 23$:           MOV     #-1,$FUNC         ; INDICATE ERROR
       000560  016767  000000G 000000G                MOV     $FUNC,RQB         ; RETURN ERROR TO CALLER
       000566  032767  000000G 000000G 25$:           BIT     #VF.NTI,$VFLAG    ; IN NTINIT MODE ?
       000574  001402                                 BEQ     40$               ; NO
       000576  000167  000000G                        JMP     $RSNTI            ; RESUME NTINIT PROCESSING
       000602                                 40$:    RETURN                    ; RESTORE GCL CONTEXT
                                              ;
                                              ; ERROR CONDITIONS
                                              ;
       000604                                 101$:   ERRPT$  UNS1,$EROUT       ; UNSUPPORTED COMMAND
       000614                                         RETURN

       000616                                         ERMSG$  <Function not implemented>
       000646                                         ERBLK$  UNS1
```

```
    716                                                  .SBTTL  LDLIN - LOAD A LINE
    717                                          ;+
    718                                          ; LDLIN - LOAD THE LINE
    719                                          ;
    720                                          ; INPUTS:
    721                                          ;        $SLTA=SLT ADDRESS
    722                                          ;        $SLN=SYSTEM LINE NUMBER
    723                                          ;        $$CSR=CSR ADDRESS
    724                                          ;        $$PRI=INTERRUPT PRIORITY (1 THRU 7)
    725                                          ;        $$VECT=INTERRUPT VECTOR ADDRESS
    726                                          ;        $$MTP=STATION COUNT (MULTIPOINT ONLY)
    727                                          ;        $$SNUM=STATION NUMBER TABLE (MULTIPOINT ONLY)
    728                                          ;
    729                                          ; OUTPUTS:
    730                                          ;        C-BIT=SUCCESS/FAILURE
    731                                          ;-
    732
    733 002320                                   LDLIN:
    734 002320  016701  000000G                          MOV     $DLCPV,R1       ; GET DLC PDV ADDRESS
    735 002324  001422                                   BEQ     20$             ; IF EQ, COMBINATION DDM/DLC
    736 002326                                           CALL    AIC             ; IS DLC ALREADY IN CORE?
    737 002332  001014                                   BNE     10$             ; IF NE, YES
    738 002334  012767  000000C 000000G                  MOV     #MS.INC!MS.EXT,$MISS ; LOOK FOR PROCESS EXTENDS IN CETAB
    739 002342                                           CALL    $NLCFG          ; ...
    740 002346  103445                                   BCS     50$             ; IF CS, FAILED
    741 002350                                           CALL    $XFER           ; TRANSFER DLC INTO CORE
    742 002354  103442                                   BCS     50$             ; IF CS, FAILURE (NOTHING TO CLEAN UP)
    743 002356  052767  000000G 000000G                  BIS     #FL.PX1,$FLAGS  ; INDICATE SUCCESSFUL DLC TRANSFER
    744 002364                                   10$:    CALL    $TMP            ; SETUP DLC LINE TABLE
    745 002370  103435                                   BCS     101$            ; IF CS, FAILURE
    746 002372  016701  000000G                  20$:    MOV     $DDMPV,R1       ; GET DDM PDV ADDRESS
    747 002376                                           CALL    AIC             ; IS DDM ALREADY IN CORE?
    748 002402  001014                                   BNE     30$             ; IF NE, YES
    749 002404  012767  000000C 000000G                  MOV     #MS.INC!MS.EXT,$MISS ; LOOK FOR PROCESS EXTENDS IN CETAB
    750 002412                                           CALL    $NLCFG          ; ...
    751 002416  103422                                   BCS     101$            ; IF CS, FAILED
    752 002420                                           CALL    $XFER           ; TRANSFER DDM INTO CORE
    753 002424  103417                                   BCS     101$            ; IF CS, FAILURE
    754 002426  052767  000000G 000000G                  BIS     #FL.PX2,$FLAGS  ; INDICATE SUCCESSFUL DDM TRANSFER
    755 002434                                   30$:
    756                                                  .IF DF  R$$MPL                                          ;[MP01]
    757                                                  CALL    $CTSYM          ; FIND CTB ADDRESS                      ;[MP01]
    758                                                  .ENDC                                                   ;[MP01]
    759
    760 002434                                           CALL    $TMP            ; SETUP DDM LINE TABLE
    761 002440  103411                                   BCS     101$            ; IF CS, FAILURE
    762
    763 002442                                   40$:    CALL    $CSRLD          ; SET UP FOR CSR ADDRESS CHECKING FOR NTINIT
    764 002446  103406                                   BCS     101$
    765 002450  005767  000000G                          TST     $$HTIM          ; HELLO AND LISTEN TIMERS SPECIFIED?
    766 002454  001402                                   BEQ     50$             ; BR IF NO
    767 002456                                           CALL    $ALTLC          ; ALLOCATE TRANSPORT LINE COUNTERS
    768
    769 002462                                   50$:    RETURN
    770
    771                                          ;
    772                                          ; ERROR CONDITION
```

```
**FILE**ID**VNSD






VV      VV   NN      NN     SSSSSSSS   DDDDDDDD
VV      VV   NN      NN     SSSSSSSS   DDDDDDDD
VV      VV   NN      NN   SS           DD      DD
VV      VV   NN      NN   SS           DD      DD
VV      VV   NNNN    NN   SS           DD      DD
VV      VV   NNNN    NN   SS           DD      DD
VV      VV   NN  NN  NN     SSSSSS     DD      DD
VV      VV   NN  NN  NN     SSSSSS     DD      DD
VV      VV   NN    NNNN           SS   DD      DD
VV      VV   NN    NNNN           SS   DD      DD
  VV  VV     NN      NN           SS   DD      DD        ....
  VV  VV     NN      NN           SS   DD      DD        ....
    VV       NN      NN   SSSSSSSS     DDDDDDDD          ....
    VV       NN      NN   SSSSSSSS     DDDDDDDD          ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```


                                ;+
                                ; DLL DATA
                                ;-

000110                          SERBEG::
000110                          SERFLG::.BLKW   1               ; SER$DF PARSING FLAG
000112                          SERNOD::.BLKW   1               ; NODE ADDRESS
000114                          SERHST::.BLKW   1               ; HOST ADDRESS
000116                          SERCIR::.BLKW   1               ; SLN/STA PAIR
000120                          SERDEV::.BLKW   1               ; DEVICE NAME (RAD50)

000122                          SERPAS::.BLKB   8.              ; PASSWORD
000132                          ENDPAS::
000132                          SERDMP::.BLKW   2               ; DUMP ADDRESS (22 BITS)
000136                          SERDCT::.BLKW   1               ; DUMP COUNT
000140                          SERADD::.BLKW   3               ; HARDWARE ADDRESS
000146                          SERTMP::.BLKW   1               ; WORK LOCATION

000150                          CIRNAM::.BLKW   2               ; CIRCUIT NAME
000154                          CIRNUM::.BLKW   1               ; CIRCUIT NUMBER
000156                          CIRMUL::.BLKW   1               ; TRIB NUMBER
000160                          DEVNM:: .BLKW   2               ; DEVICE NAME (ASCII)
000164                          HEXSTR::.BLKB   16.             ; HEX STRING BUFFER
000204                          ENDHEX::

000204                          STRSIZ::.BLKW   1               ; STRING LENGTH
000206                          UICBUF::.BLKW   6               ; UIC BUFFER
               000014           UISZ==.-UICBUF                  ; UIC BUFFER SIZE
000222                          FILBUF::.BLKB   16.             ; FILE NAME BUFFER
               000020           FLSZ==.-FILBUF                  ; FILE NAME BUFFER SIZE

000242                          DEVBUF::.BLKW   4               ; DEVICE NAME BUFFER
               000010           DVSZ==.-DEVBUF                  ; DEVICE NAME BUFFER SIZE
000252                          FILFLG::.BLKW   1               ; FIL$DF PARSING FLAG
000254                          CIRUNT::.BLKW   1               ; UNIT NUMBER OF CIRCUIT
000256                          SERCUR::.BLKW   1               ; CURRENT SERVICE BLOCK UNMAPPED ADDRESS
000260                          SEROFF::.BLKW   1               ; FNB POINTER OFFSET INTO SERVICE BLOCK

000262                          SERSLT::.BLKW   1               ; SLT ADDRESS OF ASSOCIATED CIRCUIT
000264                          FNBLN:: .BLKW   1               ; FNB LENGTH
000266                          SERFIN::
000266                          FNBUFF::.BLKB   36.             ; LOCAL COPY OFF CURRENT
               000044           FNBSZ==.-FNBUFF                 ; SIZE OF LOCAL FNB BUFFER
000332         000266'          STRAD::FNBUFF                   ; ADDRESS FOR START OF BUFFER
000334         000266'          STRPT:: FNBUFF                  ; CURRENT BUFFER LOCATION
               000006           FNBEXT  = 6                     ; FILE NAME BLOCK EXT.
               000003           FIELDC  = 3                     ; FNB FIELD COUNT
               000003           EXTN    = 3                     ; FILENAME EXTENSION MAX SIZE

                                ;
                                ; EQUATED SYMBOLS
                                ;
               001777           MAXCST= 1023.                   ; MAXIMUM COST ALLOWED
               000037           MAXHOP= 31.                     ; MAXIMUM NUMBER OF HOPS
               000100           LET    = 100                    ; HEX LETTER
```

```
    754                                                     .SBTTL ALLOCATE CCBS ( AND EXTRAS FOR RDBS )
    755
    756                                             ;+
    757                                             ;       **-$CCBS        - ALLOCATE NORMAL CCBS AND RDB'S CCBS
    758                                             ;
    759                                             ; INPUTS:
    760                                             ;
    761                                             ;       $CCBNM  - NUMBER OF CCBS TO ALLOCATE
    762                                             ;       $CCBSZ  - SIZE OF EACH CCB IN BYTES
    763                                             ;       $RDBNM  - NUMBER OF RDB TO ALLOCATE
    764                                             ;
    765                                             ; OUTPUTS:
    766                                             ;
    767                                             ;       ALL REGISTERS PRESERVED
    768                                             ; C-BIT CLEAR
    769                                             ;       $CCBLH  - LISTHEAD OF CCBS CHAINED WITH $CCBNM+$RDBNM CCBS
    770                                             ; C-BIT SET
    771                                             ;       $CCBLH  - LISTHEAD OF CCBS CHAIND WITH 0 OR MORE CCBS BUT
    772                                             ;                 LESS THAN $CCBNM+$RDBNM CCBS.
    773                                             ;-
    774
    775 002554  004267  000000G                     $CCBS:: JSR     R2,$SAVVR               ; SAVE R0,R1, AND R2
    776 002560  016702  000000G                             MOV     $CCBNM,R2               ; NUMBER OF CCBS TO ALLOCATE
    777
    778 002564  016701  000000G                     10$:    MOV     $CCBSZ,R1               ; NUMBER OF BYTES/CCB
    779
    780                                                     .IF     DF,R$$MPL
    781                                                     TST     .MPRO                   ; MULTIPROCESSOR SYSTEM?
    782                                                     BMI     12$                     ; BR IF NO
    783                                                     ADD     #2,R1                   ; NEED AN EXTRA WORD
    784                                             12$:    .ENDC
    785
    786 002570                                              CALL    $ALL16                  ; ATTEMPT ALLOCATION FROM SCOM
    787 002574  103422                                      BCS     20$                     ; RESOURCE ERROR
    788
    789                                                     .IF     DF,R$$MPL
    790                                                     TST     .MPRO                   ; MULTIPROCESSOR SYSTEM?
    791                                                     BMI     14$                     ; BR IF NO
    792                                                     ADD     #2,R0                   ; EXTRA WORD IS BEFORE THE LINK WORD
    793                                             14$:    .ENDC
    794
    795 002576  016767  000000G 000000G                     MOV     $CCBLH,$BFR             ; ADD
    796 002604  112767  000002  000003G                     MOVB    #CB.CCB,$BFR+C.BID      ;  CCB
    797 002612                                              PUTRC   R0,$CCBSZ               ;   TO
    798 002630  010067  000000G                             MOV     R0,$CCBLH               ;    QUEUE
    799 002634  005302                                      DEC     R2                      ; LOOP TILL
    800 002636  001352                                      BNE     10$                     ;  DONE.
    801 002640  000241                                      CLC                             ; SIGNAL SUCCESS
    802 002642                                      20$:    RETURN
```

```

003632                          ISTAT$  IBUF5,KBUF5

003632                          STATE$  BRADF
003632                           TRAN$  %SLT$DF%
003632                          STATE$
003632                           TRAN$   $RAD50
003632                          STATE$
003632                           TRAN$   <',>
003632                          STATE$
003632                           TRAN$   $RAD50
003632                          STATE$
003632                           TRAN$   <',>
003632                          STATE$
003632                           TRAN$   $RAD50
003632                          STATE$
003632                           TRAN$   <',>
003632                          STATE$
003632                           TRAN$   <',>,SLTDF2
003632                           TRAN$   $LAMDA
003632                          STATE$  SLTDF3
003632                           TRAN$   %LF.MFL%,SLTDF1
003632                           TRAN$   %LF.ENA%,SLTDF1
003632                           TRAN$   %LF.SER%,SLTDF1
003632                           TRAN$   $RAD50
003632                          STATE$  SLTDF1
003632                           TRAN$   '!,SLTDF3
003632                           TRAN$   <',>
003632                          STATE$  SLTDF2
003632                           TRAN$   $NUMBR
003632                          STATE$
003632                           TRAN$   <',>
003632                          STATE$
003632                           TRAN$   $NUMBR
003632                          STATE$
003632                           TRAN$   $EOS,$EXIT
003632                           TRAN$   <',>
003632                          STATE$
003632                           TRAN$   <',>,SCTIMM
003632                           TRAN$   $STRNG
003632                          STATE$
003632                           TRAN$   $EOS,$EXIT
003632                           TRAN$   <',>
003632                          STATE$  SCTIMM
003632                           TRAN$  $NUMBR,SCHEL1
003632                           TRAN$  <',>,SCHELL
003632                          STATE$  SCHEL1
003632                           TRAN$  $EOS,$EXIT
003632                           TRAN$  <',>
003632                          STATE$  SCHELL
003632                           TRAN$  $NUMBR,BRADJJ
003632                           TRAN$  <',>
003632                          STATE$  BRADJJ
003632                           TRAN$  <',>,BRADJ2
003632                           TRAN$  $EOS,$EXIT
003632                          STATE$  BRADJ2
003632                           TRAN$  $NUMBR,$EXIT,BRADJT
```

```
005766                                          CALL    $CLDEQ          ; CLOSE CONFIG FILE
005772                                          NTIERR  424             ; ROU$DF SYNTAX ERROR
                                        ;
                                        ; BUF$DF
                                        ;
006002  005767  000000G                 18$:    TST     CFSFLG          ; HAS BUF$DF ALREADY BEEN PROCESSED?
006006  100024                                  BPL     30$             ; IF PL, YES
006010  012702  000002'                         MOV     #KBUF,R2        ; LOOK FOR BUF$DF
006014  012705  000044'                         MOV     #IBUF,R5        ; ...
006020                                          CALL    PARSE           ; PARSE LINE
006024  103404                                  BCS     20$             ; IF CS, NOPE
006026  042767  100001  000000G                 BIC     #100001,CFSFLG  ; INDICATE BUF$DF SEEN
006034  001444                                  BEQ     50$             ; IF ZERO, ALL DONE
006036  105767  000000G                 20$:    TSTB    CFSFLG          ; WAS "BUF$DF" ON THIS LINE
006042  001406                                  BEQ     30$             ; IF EQ, NO
006044                                          CALL    $CLDEQ          ; CLOSE CONFIG FILE
006050                                          NTIERR  415             ; BUF$DF SYNTAX ERROR
006060  032767  040000  000000G         30$:    BIT     #40000,CFSFLG   ; HAS PAR$DF ALREADY BEEN PROCESSED?
006066  001661                                  BEQ     10$             ; IF EQ, YES
006070  005067  000000G                         CLR     NONMAP          ; INIT NON MAPPED POOL SIZE
006074  012702  000004'                         MOV     #KBUF2,R2       ; LOOK FOR PAR$DF
006100  012705  000124'                         MOV     #IBUF2,R5
006104                                          CALL    PARSE
006110  103405                                  BCS     40$             ; IF CS, NOPE
006112  042767  040001  000000G                 BIC     #40001,CFSFLG   ; INDICATE PAR$DF SEEN
006120  001412                                  BEQ     50$             ; IF ZERO, ALL DONE
006122  000643                                  BR      10$             ; BR IF MORE TO LOOK FOR
006124  105767  000000G                 40$:    TSTB    CFSFLG          ; WAS "PAR$DF" ON THIS LINE?
006130  001640                                  BEQ     10$             ; IF EQ, NO
006132                                          CALL    $CLDEQ          ; CLOSE CONFIG FILE
006136                                          NTIERR  405             ; SYNTAX ERROR IN PAR$DF MACRO
006146  066767  000002G 000000G         50$:    ADD     NTPSIZ+2,NTPSIZ ; ADD IN FUDGE FACTOR (CLEARS C-BIT)
006154  103013                          60$:    BCC     70$             ; BR IF NO ERROR
006156  006016                                  ROR     (SP)            ; SAVE C-BIT ON STACK
006160  022767  020000  000000G                 CMP     #20000,CFSFLG   ; ONLY ROU$DF MISSING?
006166  001007                                  BNE     80$             ; BR IF NO
006170  066767  000002G 000000G                 ADD     NTPSIZ+2,NTPSIZ ; ADD IN FUDGE FACTOR
006176  042716  100000                          BIC     #100000,(SP)    ; ELSE OK - CLEAR CARRY
006202  000401                                  BR      80$             ; FINISH UP
006204  006016                          70$:    ROR     (SP)            ; SAVE C-BIT ON STACK
006206                                  80$:    CALL    $CLDEQ          ; CLOSE FILE

006212  005767  000000G                         TST     SBFLG           ; SERVER BASE POOL INDICATED
006216  001403                                  BEQ     90$             ; IF NOT - BRANCH
006220  166767  000002G 000000G                 SUB     NTPSIZ+2,NTPSIZ ; REDUCE POOL.. SIZE BY THE TOTAL RDB/SDB SIZE

006226  006316                          90$:    ASL     (SP)            ; RESTORE C-BIT
006230                                          RETURN

                                        ;
                                        ; PARSE A LINE
                                        ;
006232  005001                          PARSE:  CLR     R1              ; REQUIRE FULL LENGTH KEYWORDS
006234  016703  000000G                         MOV     LINSIZ,R3       ; RECORD LENGTH
006240  016704  000000G                         MOV     $CLBUF,R4       ; RECORD SIZE
006244  010046                                  MOV     R0,-(SP)        ; SAVE FDB ADDRESS
006246                                          CALL    .TPARS          ; PARSE THE LINE
```

```
                                                   .SBTTL  REM$DF PARSE TABLES
                                          ;+
                                          ;        STATE TRANSITIONS FOR THE 'REM$DF' MACRO
                                          ;-

        006324                                     STATE$  REMDF
        006324                                      TRAN$  LBRA

        006324                                     STATE$
        006324                                      TRAN$  $STRNG,,.RNAME

        006324                                     STATE$
        006324                                      TRAN$  RBRA

        006324                                     STATE$
        006324                                      TRAN$  <',>

        006324                                     STATE$
        006324                                      TRAN$  $NUMBR,,.RARE

        006324                                     STATE$
        006324                                      TRAN$  <',>

        006324                                     STATE$
        006324                                      TRAN$  $NUMBR,$EXIT,.RNUM
```

```
                                                .SBTTL  ACTION ROUTINE: BUF$DF ARGUMENT SEEN
                                        ;+
                                        ;       **-.BUFNM-BUF$DF ARGUMENT SEEN
                                        ;
                                        ;       THE NEXT ARGUMENT OF THE BUF$DF MACRO HAS BEEN SEEN, MOVE THE
                                        ;       VALUE INTO THE CORRECT POSITION IN THE COMMEXEC DATABASE.
                                        ;-
                                        ;
006356  005367  000000G                 .BUFNM: DEC     DFCNT           ; HAVE WE SEEN ALL THE ARGUMENTS
006362  002407                                  BLT     10$             ; YES ... IGNORE THIS ONE
006364  017700  000000G                         MOV     @DFPTR,R0       ; GET NEXT POINTER FROM TABLE
006370  016710  000000G                         MOV     .PNUMB,(R0)     ; MOVE ARGUMENT TO COMMEXEC DATABASE IMAGE
006374  062767  000002  000000G                 ADD     #2,DFPTR        ; MOVE ON ARGUMENT POINTER
006402                                  10$:    RETURN
```

```
   2901                                            .SBTTL  ACTION ROUTINE: RIGHT ANGLE BRACKET SEEN
   2902                                    ;+
   2903                                    ;       **-.STRNG-RIGHT ANGLE BRACKET (END OF STRING) SEEN
   2904                                    ;
   2905                                    ;       THE TERMINATING RIGHT ANGLE BRACKET HAS BEEN SEEN ON THE 'NOD$DF' MACRO.
   2906                                    ;-
   2907
   2908 007706  016700  000324          .STRNG: MOV     STRPTR,R0               ; GET FIRST TEXT CHARACTER ADDRESS
   2909 007712  016701  000000G                 MOV     .PSTPT,R1               ; GET RIGHT ANGLE BRACKET ADDRESS
   2910 007716  005301                          DEC     R1                      ;
   2911 007720  160001                          SUB     R0,R1                   ; CALCULATE LENGTH OF TEXT IN R1
   2912 007722  016702  000312                  MOV     STRMAX,R2               ; CHECK AGAINST MAXIMUM TEXT LENGTH
   2913 007726  160102                          SUB     R1,R2
   2914 007730  100537                          BMI     101$                    ; IF MI, TEXT IS TOO LONG!
   2915 007732  010246                          MOV     R2,-(SP)                ; SAVE BLANK FILL COUNT
   2916 007734  022767  000040  000276          CMP     #32.,STRMAX             ; ID STRING ?
   2917 007742  001043                          BNE     5$                      ; IF NO - BRANCH
   2918
   2919 007744                                  SWTCHI  #$HMBUF                 ; SET INPUT BUFFER
   2920 007752                                  GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
   2921 007772  016700  000020G                 MOV     $HMBUF+D$LID,R0         ; GET ID STRING ADDRESS
   2922 007776                                  SWTCHI                          ; RESET BUFFER
   2923 010004                                  SWTCHO                          ; RESET BUFFER
   2924
   2925 010012                                  CEACC$  R0                      ; MAP TO POOL..
   2926 010022                                  GETAD   R0,#34.                 ; READ POOL.. BLOCK FOR ID STRING
   2927 010040  016700  000172                  MOV     STRPTR,R0               ; GET FIRST TEXT CHAR ADDRESS
   2928 010044  012767  000000G 000170          MOV     #$BFR,STRADR            ; GET ADDRESS OF OUTPUT BUFFER
   2929
   2930 010052  016702  000164          5$:     MOV     STRADR,R2               ; PLACE TO PUT THE TEXT
   2931 010056  001002                          BNE     10$                     ; IF VALID ADDRESS - BRANCH
   2932 010060  005726                          TST     (SP)+                   ; POP BLONK FILL COUNT
   2933 010062  000421                          BR      30$                     ; RETURN
   2934
   2935 010064  022767  000040  000146  10$:    CMP     #32.,STRMAX             ; IS THIS THE NODE ID STRING?
   2936 010072  001004                          BNE     15$                     ; IF NE, NO
   2937 010074  010122                          MOV     R1,(R2)+                ; STORE COUNT IN FIRST WORD
   2938 010076  001002                          BNE     15$                     ; IF NOT NULL NODE ID BRANCH
   2939 010100  005726                          TST     (SP)+                   ; IF NULLNODE ID CLEAN STACK
   2940 010102  000411                          BR      30$
   2941
   2942 010104  112022                  15$:    MOVB    (R0)+,(R2)+             ; COPY THE TEXT BETWEEN THE ANGLE BRACKETS
   2943 010106  005301                          DEC     R1
   2944 010110  001375                          BNE     15$
   2945 010112  012601                          MOV     (SP)+,R1                ; NOW BLANK  ILL TO MAXIMUM LENGTH
   2946 010114  001404                          BEQ     30$
   2947 010116  112722  000040          20$:    MOVB    #' ,(R2)+
   2948 010122  005301                          DEC     R1
   2949 010124  001374                          BNE     20$
   2950
   2951 010126                          30$:
   2952 010126  022767  000040  000104          CMP     #32.,STRMAX             ; ID STRING ?
   2953 010134  001016                          BNE     35$                     ; IF NO - BRANCH
   2954 010136  016700  000020G                 MOV     $HMBUF+D$LID,R0         ; GET UNMAPPED ADDRESS
   2955 010142                                  CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
   2956 010152                                  PUTAD   R0,#34.                 ; UPDATE ID STRING
   2957 010170                                  RETURN
```

```
                                        .SBTTL  ACTION ROUTINE: BROADCAST ENDNODES

                                ;
                                ; .RBEA - BROADCAST ENDNODE ADJACENCIES ACTION ROUTINE
                                ;

012250                          .RBEA:
012250                                  GETRV   $DECPT,#D$END   ; DECNET HOME BLOCK

012270  005767  000060G                 TST     $BFR+D$NN       ; ROUTING NODE ?
012274  001004                          BNE     10$             ; IF YES - BRANCH
012276  005767  000000G                 TST     .PNUMB          ; IF NO - VALUE MUST BE ZERO
012302  001405                          BEQ     20$             ; IF LEGAL - BRANCH
012304  000412                          BR      30$             ; ELSE - REPORT ERROR

012306  022767  002000  000000G 10$:    CMP     #1024.,.PNUMB   ; CHECK HIGH LIMIT
012314  103406                          BLO     30$             ; IF OUT OF RANGE - BRANCH

012316  016767  000000G 000056G 20$:    MOV     .PNUMB,$BFR+D$NBEA ; COPY NUMBER OF BROADCAST ENDNODES

012324                                  PUTRC                   ; WRITE BACK DECNET HOME BLOCK

012330                                  RETURN

012332  062716  000002          30$:    ADD     #2,(SP)         ; ILLEGAL ADJACENT BROADCAST ENDNODE VALUE
012336                                  RETURN


                                ;
                                ; .RAMXH - AREA MAX HOPS
                                ;

012340  022767  000036  000000G .RAMXH: CMP     #30.,.PNUMB     ; CHECK HIGH LIMIT
012346  103416                          BLO     30$             ; IF OUT OF RANGE - BRANCH

012350                                  GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK

012370  016767  000000G 000074G         MOV     .PNUMB,$BFR+D$AMXH ; COPY AREA MAX HOPS

012376                                  PUTRC                   ; WRITE BACK DECNET HOME BLOCK

012402                                  RETURN

012404  062716  000002          30$:    ADD     #2,(SP)         ; ILLEGAL AREA MAX HOPS VALUE
012410                                  RETURN

```

```
                                           .SBTTL  FINREM - FIND PLACE FOR A REMOTE NODE BLOCK
                                   ;+
                                   ; *** - FINREM - FIND A PLACE FOR A REMOTE NODE BLOCK
                                   ;
                                   ; INPUTS:
                                   ;       R1 = NODE ADDRESS OR CHANNEL TO BE INSERTED
                                   ;
                                   ; OUTPUTS:
                                   ;       R0 = PRESERVED
                                   ;       R1 = UNMAPPED ADDRESS OF PREVIOUS REMOTE BLOCK
                                   ;       R2 = DESTROYED
                                   ;-

013732                             FINREM:
013732                                     SWTCHI  #$HMBUF                 ; USE TEMPORARY BUFFER
013740                                     GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
013760                                     SWTCHI                          ; RESET DEFAULT BUFFER

013766  012702   000004G                   MOV     #$HMBUF+D$RNN+2,R2      ; GET START OF REMOTES LIST
013772  011267   000000G                   MOV     (R2),$BFR               ; GET FIRST IN LIST
013776  010146                             MOV     R1,-(SP)                ; SAVE NODE ADDRESS ON STACK
014000  010267   000004'                   MOV     R2,CURUNM               ; SAVE UNMAPPED ADDRESS OF CURRENT
014004  016767   000004' 000002'   10$:    MOV     CURUNM,PRVUNM           ; SAVE UNMAPPED ADDRESS OF PREVIOUS
014012  016767   000000G 000004'           MOV     $BFR,CURUNM             ; GET NEXT REMOTE IN LIST
014020  001005                             BNE     15$                     ; IF NE, CONTINUE
014022  010167   000000G                   MOV     R1,$LSTAD               ; UPDATE NEW HIGH NODE ADDRESS
014026  010067   000000G                   MOV     R0,$LSTRM               ; UPDATE NEW LAST NODE BLOCK POINTER
014032  000417                             BR      30$                     ; CONTINUE

014034                             15$:    CEACC$  CURUNM,R2               ; CONVERT TO UNMAPPED ADDRESS
014046                             20$:    GETAD   R2,#R.LEN               ; READ REMOTE IN INTERNAL BUFFER
014064  021667   000010G                   CMP     (SP),$BFR+R.ADD         ; DOES REMOTE BLOCK GO HERE ?
014070  101345                             BHI     10$                     ; IF HI, NO
014072  016701   000002'           30$:    MOV     PRVUNM,R1               ; RETRIEVE UNMAPPED ADDRESS OF PREVIOUS
014076  005726                             TST     (SP)+                   ; CLEAN UP THE STACK
014100                                     RETURN

```

```

                                        ;+
                                        ; SRHFNB - SEARCH FOR FILE NAME BLOCK
                                        ;
                                        ; INPUTS -
                                        ;               FNBUFF CONTAINS FILE DESCRIPTOR INFORMATION
                                        ;
                                        ; OUTPUTS -
                                        ;               C CLEAR - FILE NAME BLOCK FOUND
                                        ;                       - R0 CONTAINS UNMAPPED BLOCK ADDRESS
                                        ;
                                        ;               C SET   - FILE NAME BLOCK NOT FOUND
                                        ;-

 015552                                 SRHFNB: SAVRG   <R3,R4,R5>              ; SAVE REGS
 015560                                         SWTCHI  #$HMBUF         ; SWITCH INPUT BUFFER
 015566                                         GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK
 015606                                         SWTCHI                  ; RESET INPUT BUFFER

 015614 012700  000034G                         MOV     #$HMBUF+D$FNB,R0 ; GET LISTHEAD
 015620 010003                                  MOV     R0,R3           ; SAVE LISTHEAD
 015622 011300                          5$:     MOV     (R3),R0         ; GET ADDRESS OF FIRST BLOCK
 015624 001005                                  BNE     10$             ; IF NOT END OF LIST BRANCH
 015626 000261                                  SEC                     ; SET C-BIT
 015630                                         RESRG   <R5,R4,R3>      ; RESTORE REGS
 015636                                         RETURN

 015640 010002                          10$:    MOV     R0,R2           ; SAVE UNMAPPED BLOCK ADDRESS

 015642                                         CEACC$  R0              ; MAP TO BLOCK
 015652                                         GETAD   R0,#FNBSZ       ; READ BLOCK
 015670 012703  000000G                         MOV     #$BFR,R3        ; GET START OF BLOCK
 015674 012700  000000G                         MOV     #$BFR,R0        ; ...

 015700 012701  000266'                         MOV     #FNBUFF,R1      ; COPY ADDRESS OF LOCAL FILE BLOCK
 015704 062700  000003                          ADD     #F.USE+1,R0     ; POINT TO FILESPEC DATA OF CURRENT BLOCK
 015710 062701  000003                          ADD     #F.USE+1,R1     ; POINT TO FILESPACE DATA OF LOCAL BLOCK

 015714 012705  000003                          MOV     #FIELDC,R5      ; SET UP FIELD COUNT
 015720                                         CALL    CMPSTR          ; CHECK FOR FIELD EQUALITY
 015724 103736                                  BCS     5$              ; IF NO MATCH - BRANCH

 015726 010200                                  MOV     R2,R0           ; IF MATCH - COPY UNMAPPED ADDRESS
 015730                                         RESRG   <R5,R4,R3>              ; RESTORE REGS

 015736                                         RETURN

```

```
                                                 .SBTTL  PERFORM 'MARKED FOR LOAD' SCAN
                                         ;+
                                         ;       **-$MFL-PERFORM 'MARKED FOR LOAD' SCAN
                                         ;
                                         ;       SCAN THE COMMEXEC DATABASE TO DISCOVER WHICH PROCESSES AND
                                         ;       LINES HAVE THE 'MARKED FOR LOAD' ATTRIBUTE. REQUEST THE NETWORK
                                         ;       LOADER TO LOAD ALL LINES AND PROCESSES SO MARKED.
                                         ;
                                         ;-

 017360                                  $MFL::  CALL    $SAVAL          ; SAVE ALL REGISTERS$_N ENTRY

                                         ; LOAD PROCESSES
                                         ;
 017364  016701  000000G                         MOV     $PDVTA,R1       ; POINT
 017370  016700  000000G                         MOV     $PDVNM,R0       ; GET # OF ENTRIES
 017374  012102                          10$:    MOV     (R1)+,R2        ; GET A PDV POINTER
 017376  001411                                  BEQ     20$             ; IGNORE ZERO ENTRIES
 017400  005762  000012                          TST     Z.PCB(R2)       ; IS THE PROCESS ALREADY LOADED?
 017404  001006                                  BNE     20$             ; YES ... DON'T LOAD IT AGAIN
 017406  132762  000010  000010                  BITB    #ZF.MFL,Z.FLG(R2) ; IS PROCESS 'MARKED FOR LOAD' ?
 017414  001402                                  BEQ     20$             ; IF EQ, NO
 017416                                          CALL    LDPRO           ; ELSE, LOAD THIS PROCESS
 017422  005300                          20$:    DEC     R0
 017424  001363                                  BNE     10$             ; LOOP TILL ALL PDV'S HAVE BEEN SCANNED

 017426  016700  000000G                         MOV     $SLTMA,R0       ; POINT TO THE SLT MAPPING TABLE
 017432  016746  000000G                         MOV     $SLTNM,-(SP)    ; NUMBER OF SYSTEM LINES
 017436  001425                                  BEQ     45$             ; IF EQ, NONE
 017440  012001                          30$:    MOV     (R0)+,R1        ; GET NEXT SLT
 017442  032761  004000  000000                  BIT     #LF.MFL,L.FLG(R1) ; LINE 'MARKED FOR LOAD' ?
 017450  001416                                  BEQ     40$             ; IF EQ, NO
 017452  116102  000002                          MOVB    L.DDM(R1),R2    ; GET DDM PDV INDEX
 017456  066702  000000G                         ADD     $PDVTA,R2       ; POINT INTO PDV VECTOR TABLE
 017462  011202                                  MOV     (R2),R2         ; GET PDV ADDRESS
 017464  032762  002000  000010                  BIT     #ZF.PSE,Z.FLG(R2) ; IS THIS A DLM CIRCUIT?
 017472  001403                                  BEQ     35$             ; BR IF NO
 017474                                          CALL    LDDLM           ; ELSE, REQUEST NETLOADER TO LOAD DLM CIRCUIT
 017500  000402                                  BR      40$             ; CONTINUE

 017502                                  35$:    CALL    LDLINE          ; ELSE, REQUEST NETLOADER TO LOAD THE LINE

 017506  005316                          40$:    DEC     (SP)            ; ONE LESS LINE
 017510  003353                                  BGT     30$             ; LOOP TILL DONE

 017512  005726                          45$:    TST     (SP)+           ; CLEAN UP STACK
 017514                                          RETURN
```

```
Symbol table

LX.LIN= 000006          OP$CCS= 000004          OP$X5S= 000004          S2.BEL= 020000          S5.XON= 000010
LX.LOG= 000021          OP$CIE= 004000          OP$X9S= 000010          S2.BRQ= 000020          S6.EIO= 000400
LX.NOD= 000015          OP$CON= 000400          OTSKTL  001347R     002 S2.CR = 002000          S6.RDI= 100000
LX.OBJ= 000023          OP$COP= 000001          PARTAB= ****** GX       S2.CTO= 040000          S6.RLU= 001000
LX.PRO= 000005          OP$COS= 000100          PASWTL  000637R     002 S2.CTS= 100000          TRIBTL  000737R     002
L$$ASG= 000000          OP$DST= 000200          PCKBCD  006204R         S2.ELF= 001000          TY$CHA= 000400
L$$DRV= 000000          OP$DTE= 000040          PF.XDF= ****** GX       S2.FLF= 000400          TY$EVE= 001000
L$$P11= 000001          OP$DUP= 000002          PNAMTL  000172R     002 S2.IRQ= 000200          TY$STA= 002000
L$$11R= 000000          OP$EST= 000100          PRNMTL  000321R     002 S2.OBF= 004000          TY$SUM= 000000
L.DDM = ****** GX       OP$EVE= 010000          PSWFTL  001604R     002 S2.ORQ= 000100          T$$KMG= 000000
L.FLG = ****** GX       OP$FIL= 000200          P$$P45= 000000          S2.PCU= 010000          T$$MIN= 000000
MARKER  000104R     002 OP$GRP= 000100          P$$WRD= 000000          S2.RCL= 000001          UC.ALG= 000200
MAXID = 000020          OP$HOS= 000001          QF$ACT= 000001          S2.SRQ= 000040          UC.ATT= 000010
MOVE    006522R         OP$HTM= 000002          QF$ALL= 000002          S2.WAL= 000010          UC.KIL= 000004
MSTRIB  001744R     002 OP$INA= 004000          QF$KNO= 000004          S2.WRA= 000006          UC.LGH= 000003
MSUNUM  001677R     002 OP$INC= 001000          QF$LOO= 000010          S2.WRB= 000002          UC.NPR= 000100
MXADDR= 001777          OP$KDS= 002000          Q$$OPT= 000010          S3.ACR= 000001          UC.PWF= 000020
MXALIN= 000006          OP$KDT= 000400          Q.IOAE= 000012          S3.CTC= 000004          UC.QUE= 000040
MXAREA= 000077          OP$KEV= 001000          Q.IOEF= 000006          S3.FDX= 000200          UD.UNS= 000000
MXCN64  000667R     002 OP$KGR= 001000          Q.IOFN= 000002          S3.ICE= 001000          UD.160= 000004
MXCOP = 000077          OP$KNT= 010000          Q.IOLU= 000004          S3.MHE= 000400          UD.200= 000001
MXCST = 000031          OP$KSK= 002000          Q.IOPL= 000014          S3.NEC= 000020          UD.556= 000002
MXDSTN= 000006          OP$LCT= 000200          Q.IOPR= 000007          S3.PPT= 020000          UD.625= 000005
MXDTEL= 000020          OP$LNE= 000040          Q.IOSB= 000010          S3.PTH= 004000          UD.8K = 000006
MXGRPN= 000006          OP$LOC= 000001          RAD     006052R         S3.RAL= 000010          UD.800= 000003
MXLN64  000257R     002 OP$LST= 000002          RQB   = ****** GX       S3.RES= 010000          UIDTL   001547R     002
MXNODN= 000006          OP$LIM= 000001          RQBST = ****** GX       S3.RUB= 040000          UM.CLI= 000036
MXPSCN= 000006          OP$LVL= 000010          R$$DER= 000000          S3.TAB= 000002          UM.DSB= 000200
MXSNKN= 000050          OP$MAC= 000010          R$$K11= 000001          S3.TME= 002000          UM.NBR= 000400
M$$CRB= 000124          OP$MCO= 000002          R$$SND= 000000          S3.TSY= 000040          UM.OVR= 000001
M$$CRX= 000000          OP$MDE= 000400          R$$11M= 000000          S3.8BC= 000100          UNUMTL  000421R     002
M$$FCS= 000000          OP$MLI= 000004          SCODV   000060R     002 S4.ABD= 100000          US.ABO= 000001
M$$MGE= 000000          OP$MON= 000100          SCOUN   000062R     002 S4.ANI= 000400          US.BSP= 000002
M$$NET= 000000          OP$MST= 000020          SPACE = 000040          S4.AVO= 001000          US.BSY= 000200
M$$OVR= 000000          OP$NAM= 000004          SPFILL  006536R         S4.BLK= 002000          US.FOR= 000040
NAMETL  002201R     002 OP$NET= 004000          STORID  006416R         S4.DEC= 004000          US.FRK= 000002
NODID   000072R     002 OP$NLI= 000002          STRNAM  004474R         S4.DLO= 000100          US.KPF= 000001
NODNAM  004744R         OP$NOD= 000020          SYNERR  000114R     002 S4.EDT= 010000          US.LAB= 000004
NSLINE  002016R     002 OP$OUT= 002000          SYSFDB= ****** GX       S4.HFF= 000020          US.MDE= 000002
NSTERM  001642R     002 OP$OWN= 000020          S$$WRG= 000000          S4.HFL= 000007          US.MDM= 000020
N$$ACC= 000001          OP$PAR= 000010          S$$YSZ= 007600          S4.HHT= 000040          US.MNT= 000100
N$$BUF= 000001          OP$PRI= 000010          S1.DEC= 001000          S4.HSY= 000200          US.OFL= 000001
N$$LDV= 000001          OP$RET= 010000          S1.DPR= 000400          S4.RGS= 020000          US.PUB= 000004
N$$MCP= 000001          OP$RPA= 000010          S1.DSI= 004000          S4.SFC= 040000          US.PWF= 000010
N$$MLL= 000001          OP$SCO= 000001          S1.ESC= 000002          S4.VFL= 000010          US.RED= 000002
N$$MOV= 000010          OP$SEG= 000400          S1.IBF= 002000          S5.ABP= 100000          US.SHR= 000001
N$$NCT= 000001          OP$SER= 000400          S1.IBY= 000200          S5.BCC= 020000          US.SPU= 000002
N$$PEM= 000001          OP$SIN= 000010          S1.OBY= 000100          S5.DAO= 040000          US.UMD= 000010
OCOPTL  001305R     002 OP$STA= 000004          S1.PTH= 000010          S5.HPC= 000014          US.VV = 000001
OF$INS= 000001          OP$TPA= 000020          S1.RES= 010000          S5.HPO= 000020          US.WCK= 000010
OF$LOG= 000100          OP$TRI= 000040          S1.RNE= 000020          S5.ITI= 000100          UU.ABO= 000400
OF$OFF= 000002          OP$UCS= 000020          S1.RNF= 020000          S5.OXF= 000040          UU.ATN= 000100
OF$ON = 000000          OP$USE= 000002          S1.RSP= 000004          S5.RPO= 002000          UU.AVN= 000004
OF$SMC= 000200          OP$VEC= 000040          S1.RST= 000001          S5.SW1= 000001          UU.GUS= 000010
ONAMTL  001151R     002 OP$VER= 000040          S1.TNE= 040000          S5.TMM= 000002          UU.IOS= 002000
OP$ADD= 000200          OP$XAC= 000001          S1.TSY= 000040          S5.VER= 010000          UU.ONL= 000020
OP$ALL= 000200          OP$XPR= 000002          S1.USI= 100000          S5.XOF= 000004          UU.RCT= 000002
```

```
VNCP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:44     PAGE 11
MACRO CROSS REFERENCE                                    CREF    04.00
MACRO NAME      REFERENCES

ASL$            #5-58     9-1355
CALL             6-213    6-243     6-251     7-265     7-278     7-282     7-356     7-373     7-450     7-507
                 7-549    7-561     7-636     7-638     7-781     7-804     7-838     7-852     7-865     7-898
                 7-909    7-953     7-1004    7-1065    7-1122    7-1197    7-1203    7-1233    7-1239    7-1271
                 8-1302   8-1305    8-1317    13-1473   13-1492
DHBDF$          #5-58     5-63
DIR$            #5-58     6-222     6-228     6-232     6-241     10-1374
ERROR$          #5-92     6-243     6-251     7-268     7-285     7-396     7-413     7-425     7-438     7-456
                 7-473    7-490     7-517     7-522     7-543     7-564     7-574     7-584     7-655     7-667
                 7-679    7-690     7-701     7-712     7-723     7-734     7-746     7-772     7-797     7-809
                 7-825    7-842     7-859     7-869     7-881     7-913     7-926     7-942     7-969     7-1021
                 7-1083   7-1116    7-1127    7-1148    7-1185    7-1208    7-1209    7-1221    7-1244    7-1245
                 7-1248   7-1262    7-1280
ETEXT$          #5-83     5-146     5-147     5-148     5-149     5-150     5-151     5-152     5-153     5-154
                 5-155    5-156     5-157     5-158     5-159     5-160     5-161     5-162     5-163     5-164
                 5-165    5-166     5-167     5-168     5-169     5-170     5-171     5-172     5-173     5-174
                 5-175    5-176     5-177     5-178     5-179     5-180     5-181     5-182     5-183     5-184
                 5-185    5-186     5-187     5-188     5-189
GETRV           #5-58     7-549     7-781     7-953     7-1004    7-1065    8-1317
OFF$            #5-106   #5-112
QDPB$           #5-106    5-106    #5-112     5-112
QIOW$           #5-58     5-106     5-112
RAND            #7-549    7-549    #7-781     7-781    #7-953     7-953    #7-1004    7-1004   #7-1065    7-1065
                #8-1317   8-1317
RESRG           #5-58     6-234     7-508
RETURN           6-254    7-267     7-283     7-313     7-327     7-347     7-359     7-378     7-399     7-412
                 7-424    7-437     7-455     7-472     7-489     7-514     7-526     7-542     7-553     7-563
                 7-573    7-581     7-625     7-643     7-645     7-654     7-666     7-678     7-689     7-700
                 7-711    7-722     7-733     7-745     7-754     7-770     7-795     7-808     7-824     7-832
                 7-841    7-854     7-857     7-868     7-880     7-888     7-891     7-894     7-912     7-925
                 7-940    7-968     7-995     7-1020    7-1032    7-1039    7-1052    7-1056    7-1081    7-1115
                 7-1126   7-1147    7-1164    7-1184    7-1207    7-1220    7-1243    7-1261    7-1279    8-1308
                 8-1321   9-1363    10-1369   10-1375   11-1409   12-1435   13-1494   13-1497   14-1520   15-1542
SAVRG           #5-58     6-219     7-501
SWTCHI          #5-58     7-780     7-782     7-952     7-954     7-1003    7-1005    7-1064    7-1066
UCBDF$          #5-58     5-62
VNPDF$          #5-58     5-61
```

```
**FILE**ID**VNP
```

```
                                               .SBTTL  PRCIR - SPECIAL PROCESSING FOR CIRCUITS
                                       ;+
                                       ; **-PRCIR-SPECIAL PROCESSING FOR CIRCUITS
                                       ;
                                       ; INPUTS:
                                       ;       RQB = REQUEST BLOCK
                                       ;       $QUAL = SPECIAL QUALIFERS
                                       ;
                                       ; OUTPUTS:
                                       ;       NONE.
                                       ;
                                       ;-
 000652                                PRCIR:
 000652   032767  000004  000000G              BIT     #QF$KNO,$QUAL               ; IS THIS FOR KNOWN CIRCUITS?
 000660   001004                               BNE     10$                         ; IF NE, YES
 000662   032767  003400  000000G              BIT     #WC.CNT!WC.UNT!WC.TRI,$QUAL ; ELSE, ARE THERE ANY WILD CARDS ?
 000670   001667                               BEQ     $VRQB                       ; IF EQ, NO .. NOTHING SPECIAL HERE

 000672   005004                       10$:    CLR     R4                          ; INITIALIZE SUCCESS FLAG
 000574   016705  000000G                      MOV     $SLTMA,R5                   ; GET ADDRESS OF SLT MAPPING TABLE
 000700   016703  000000G                      MOV     $SLTNM,R3                   ; GET NUMBER OF SLT'S
 000704   116746  000000G                      MOVB    $OPTON,-(SP)                ; SAVE OPTIONS WORD
 000710                                15$:    CALL    NXTLIN                      ; ELSE, GET THE NEXT LINE
 000714   103467                               BCS     60$                         ; IF CS, NO MORE
 000716   032767  000004  000000G              BIT     #QF$KNO,$QUAL               ; IS THIS FOR KNOWN CIRCUITS?
 000724   001024                               BNE     30$                         ; IF NE, YES

 000726   026167  000000G 000002G              CMP     Z.NAM(R1),RQB+LR.LIN        ; IS THIS THE RIGHT CIRCUIT?
 000734   001365                               BNE     15$                         ; IF NE, NO
                                       ;
                                       ; MATCH ON CONTROLLER
                                       ;
 000736   032767  000400  000000G              BIT     #WC.CNT,$QUAL               ; IS THIS FOR ALL CONTROLLERS ?
 000744   001004                               BNE     20$                         ; IF NE, YES
 000746   126067  000000G 000004G              CMPB    L.CTL(R0),RQB+LR.CTL        ; ELSE, IS THIS THE RIGHT CONTROLLER ?
 000754   001355                               BNE     15$                         ; IF NE, NO
                                       ;
                                       ; MATCH ON UNIT NUMBER
                                       ;
 000756   032767  001000  000000G      20$:    BIT     #WC.UNT,$QUAL                 S THIS FOR ALL UNITS ?
 000764   001004                               BNE     30$                           IF NE, YES
 000766   126067  000000G 000005G              CMPB    L.UNT(R0),RQB+LR.UNT        ; ELSE, IS THIS THE RIGHT UNIT ?
 000774   001345                               BNE     15$                         ; IF NE, NO
                                       ;
                                       ; SAVE DATA IN REQUEST BLOCK
                                       ;
 000776   005204                       30$:    INC     R4                          ; INDICATE A CIRCUIT FOUND
 001000   016167  .00000G 000002G              MOV     Z.NAM(R1),RQB+LR.LIN        ; SAVE DDM PROCESS NAME
 001006   116067  000000G 000004G              MOVB    L.CTL(R0),RQB+LR.CTL        ; SAVE CONTROLLER NUMBER
 001014   116067  000000G 000005G              MOVB    L.UNT(R0),RQB+LR.UNT        ; SAVE UNIT NUMBER
 001022   105067  000006G                      CLRB    RQB+LR.TRI                  ; ASSUME LINE IS NOT MULTIPOINT
                                       ;
                                       ; TAKE CARE OF TRIBUTARUIES
                                       ;
 001026   116002  000000G                      MOVB    L.NSTA(R0),R2               ; GET THE NUMBER OF TRIBUTARIES
 001032   001405                               BEQ     50$                         ; IF EQ, LINE IS NOT MULTIPOINT
```

```
    773
    774 002464                        101$:   CALLR   $XLIN           ; DO A PARTIAL ''CLEAR LINE''
```

```
                                        .TITLE  VNSD - VNP SYMBOL DEFINITIONS
                                        .IDENT  /V05.00/

                                ;
                                ; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                                ; TRANSFERRED.
                                ;
                                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                                ; CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ; MODULE DESCRIPTION:
                                ;
                                ;       VNP - STANDARD RSX11M/S SYMBOLS
                                ;
                                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                                ;
                                ; IDENT HISTORY:
                                ;
                                ; 1.00  27-FEB-78
                                ;       VERSION 2.0 RELEASE
                                ;
                                ; 2.00  14-DEC-79
                                ;       DECNET-11M/S V3.0
                                ;       DECNET-11M-PLUS V1.0
                                ;
                                ; 3.00  16-APR-82
                                ;       DECNET-11M V3.1
                                ;       DECNET-11M-PLUS V1.1
                                ;
                                ; 4.00  07-NOV-83
                                ;       DECNET-11M V4.0
                                ;       DECNET-11M-PLUS V2.0
                                ;
                                ; 5.00  22-JUL-85
                                ;       DECnet-11M/S V4.2
                                ;       DECnet-11M-Plus V3.0
                                ;       DECnet-Micro/RSX V1.0
                                ;
                                ;


```

```
    203
    204 000000                                 .PSECT
```

```
                                        .SBTTL  ALLOCATE RDBS

                                ;+
                                ;       **-$RDBS INITIALIZE RDB LISTHEAD
                                ;
                                ; INPUTS:
                                ;       R4      - BIAS OF BASE OF POOL
                                ;       R5      - VIRTUAL ADDRESS OF BASE OF POOL
                                ;
                                ; OUTPUTS:
                                ;       R4      - BIAS OF BASE OF NEXT FREE SEGMENT OF POOL
                                ;       R5      - VIRTUAL ADDRESS OF NEXT FREE BLOCK
                                ;       ALL OTHER REGISTERS PRESERVED
                                ;-

002644  010346                  $RDBS:: MOV     R3,-(SP)                ; SAVE R3 AND
002646  010246                          MOV     R2,-(SP)                ; SAVE R2
002650  010146                          MOV     R1,-(SP)                ; SAVE R1
002652  016701  000000G                 MOV     $RDBNM,R1               ; NUMBER OF RDBS TO ALLOCATE
002656  016703  000000G                 MOV     $RDBSZ,R3               ; NUMBER OF BYTES/RDB
002662  066703  000000G                 ADD     .CRCOV,R3               ; ADD IN THE CRC OVERHEAD
002666  005203                          INC     R3                      ; MAKE IT EVEN
002670  042703  000001                  BIC     #1,R3                   ;       ...

002674  066705  000000G         20$:    ADD     $MAXOV,R5               ; SKIP MAXIMUM DLC OVERHEAD IN RDB
002700  010467  000000G                 MOV     R4,$RBLCK               ; MAP TO BUFFER
002704  016767  000000G 000000G         MOV     $RDBLH,$BFR             ; INITIALIZE BIAS
002712  016767  000002G 000002G         MOV     $RDBLH+2,$BFR+2         ;   AND VIRTUAL ADDRESS
002720  010546                          MOV     R5,-(SP)                ; SAVE MAPPING BITS
002722  042705  160000                  BIC     #160000,R5              ; CLEAR MAPPING BITS
002726                                  PUTAD   R5,#4                   ; WRITE VALUE BACK TO NTPOOL
002744  012605                          MOV     (SP)+,R5                ; RESTORE REGISTER
002746  010567  000002G                 MOV     R5,$RDBLH+2             ; INITIALIZE LISTHEAD VIRTUAL ADDRESS
002752  010467  000000G                 MOV     R4,$RDBLH               ;    AND BIAS

002756                                  CALL    $GTBUF                  ; GET DOUBLE-WORD ADDRESS OF NEXT BUFFER
002762  005301                          DEC     R1                      ; LOOP TILL
002764  001343                          BNE     20$                     ;  DONE.
002766  012602                          MOV     (SP)+,R2                ; RESTORE R2
002770  012603                          MOV     (SP)+,R3                ; AND R3
002772  012601                          MOV     (SP)+,R1                ; AND R1
002774                                  RETURN
```

```
   1321 003632                                  STATE$
   1322
   1323
   1847 006252  012600                              MOV     (SP)+,R0           ; RESTORE FDB ADDRESS
   1848 006254                                      RETURN                     ; WITH C-BIT FROM TPARS
```

```
                                 .SBTTL  EVT$DF PARSE TABLES
                         ;+
                         ;       STATE TRANSITIONS FOR THE 'EVT$DF' MACRO
                         ;
                         ;-
006324                           STATE$  EVTDF
006324                           TRAN$   $NUMBR,,E.NUM

006324                           STATE$
006324                           TRAN$   <',>

006324                           STATE$
006324                           TRAN$   LBRA

006324                           STATE$
006324                           TRAN$   !EVTMSK

006324                           STATE$
006324                           TRAN$   <',>

006324                           STATE$
006324                           TRAN$   "CONSOLE",$EXIT,E.CON
006324                           TRAN$   "FILE",$EXIT,E.FIL
006324                           TRAN$   "MONITOR",$EXIT,E.MON

                         ;
                         ; EVTMSK SUB-EXPRESSION
                         ;
006324                           STATE$  EVT
006324                           TRAN$   $NU[illegible] , [illegible] MSK

006324                           STATE$
006324                           TRAN$   RBRA,$EXIT
006324                           TRAN$   <',>,EVTMSK
```

```
   2575                                                 .SBTTL  BUF$DF ARGUMENT CONTROL TABLE
   2576                                         ;+
   2577                                         ;       **-DFTBL-BUF$DF ARGUMENT CONTROL TABLE
   2578                                         ;
   2579                                         ;       THIS CONTROL TABLE POINTS TO THE ADDRESSES IN THE COMMEXEC
   2580                                         ;       TO RECEIVE THE ARGUMENTS OF THE BUF$DF MACRO. THE TABLE IS
   2581                                         ;       ORDERED IN THE SAME WAY AS THE ARGUMENTS OF THE BUF$DF
   2582                                         ;       MACRO.
   2583                                         ;-
   2584                                         ;
   2585 006404  000000G                         DFTBL:  .WORD   $CCBNM  ; # OF CCB'S TO ALLOCATE
   2586 006406  000000G                                 .WORD   $CCBSZ  ; SIZE OF A CCB (IN BYTES)
   2587 006410  000000G                                 .WORD   $RDBNM  ; # OF RDB'S TO ALLOCATE
   2588 006412  000000G                                 .WORD   $RDBSZ  ; SIZE OF AN RDB (IN BYTES)
   2589 006414  000000G                                 .WORD   $SDBNM  ; # OF SDB'S TO ALLOCATE
   2590 006416  000000G                                 .WORD   $SDBSZ  ; SIZE OF AN SDB (IN BYTES)
   2591 006420  000000G                                 .WORD   $RDBTH  ; RDB ALLOCATION THRESHOLD
   2592 006422  000000G                                 .WORD   $MAXOV  ; MAXIMUM PROTOCOL OVERHEAD SIZE
   2593         000010                          DFSIZ=<.-DFTBL>/2
```

```
  2958
  2959 010172                          35$:    SWTCHO  #$HMBUF
  2960 010200                                  PUTRC   $DECPT,#D$END   ; WRITE BACK DECNET HOME BLOCK
  2961 010220                                  SWTCHO
  2962
  2963 010226                                  RETURN
  2964                                 ;
  2965 010230  062716  000002          101$:   ADD     #2,(SP)         ; REJECT TPARS TRANSITION
  2966 010234                                  RETURN
  2967                                 ;
  2968                                 ; LOCAL DATA FOR ANGLE-BRACKETED STRING MANIPULATION
  2969                                 ;
  2970 010236                          STRPTR: .BLKW   1               ; ADDRESS OF FIRST TEXT BYTE
  2971 010240                          STRMAX: .BLKW   1               ; MAXIMUM TEXT STRING LENGTH
  2972 010242                          STRADR: .BLKW   1               ; PLACE IN CETAB WHERE TEXT WILL GO
```

```
                                                .SBTTL  ACTION ROUTINE: AREA MAX COST

                                        ;
                                        ; .RAMXC - AREA MAX COST ACTION ROUTINE
                                        ;

 012412  022767  002000  000000G .RAMXC: CMP     #1024.,.PNUMB   ; CHECK FOR HIGH LIMIT
 012420  103416                          BLO     30$             ; IF OUT OF RANGE - BRANCH

 012422                                  GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK

 012442  016767  000000G 000072G 20$:    MOV     .PNUMB,$BFR+D$AMXC ; COPY AREA MAX COST

 012450                                  PUTRC                   ; WRITE BACK DECNET HOME BLOCK

 012454                                  RETURN

 012456  062716  000002          30$:    ADD     #2,(SP)         ; ILLEGAL AREA MAX COST VALUE
 012462                                  RETURN

                                        ;+
                                        ; .RBTM - BROADCAST TIMER ACTION ROUTINE
                                        ;-

 012464                          .RBTM:
 012464                                  GETRV   $DECPT,#D$END   ; READ DECNET HOME BLOCK

 012504  005767  000060G                 TST     $BFR+D$NN       ; ENDNODE ?
 012510  001004                          BNE     10$             ; IF NOT - BRANCH
 012512  005767  000000G                 TST     .PNUMB          ; CHECK FOR LEGAL VALUE
 012516  001405                          BEQ     20$             ; IF OK - BRANCH
 012520  000412                          BR      30$             ; IF NOT - ERROR
 012522  022767  000377  000000G 10$:    CMP     #255.,.PNUMB    ; CHECK LIMIT
 012530  002406                          BLT     30$             ; IF ILLEGAL BRANCH

 012532  016767  000000G 000100G 20$:    MOV     .PNUMB,$BFR+D$BRTM ; STORE BROADCAST TIMER VALUE

 012540                                  PUTRC                   ; WRITE BACK DECNET HOME BLOCK

 012544                                  RETURN

 012546  062716  000002          30$:    ADD     #2,(SP)         ; ILLEGAL BROADCAST TIMER VALUE
 012552                                  RETURN

```

```

                                        .SBTTL  QUICK - FAST REMOTE NODE PROCESSING
                                ;+
                                ; QUICK - FAST REMOTE NODE PROCESSING
                                ;
                                ; INPUTS:
                                ;       R0      - POINTER TO NEW NODE BLOCK
                                ;       R1      - NODE ADDRESS TO BE INSERTED
                                ;       $LSTAD  - CURRENT HIGHEST NODE ADDRESS
                                ;       $LSTRM  - HIGHEST NODE ADDRESS POINTER
                                ;
                                ; OUTPUTS:
                                ;       C SET   - FAST SEARCH FAILURE
                                ;       C CLR   - FAST SEARCH SUCCESS
                                ;       R1      - POINTER TO NEW PREVIOUS BLOCK
                                ;       $LSTAD,$LSTRM - UPDATED
                                ;-

014102  005767  000000G         QUICK:  TST     $LSTAD          ; HIGH ADDRESS VALID ?
014106  001002                          BNE     10$             ; IF YES - BRANCH
014110  000261                  5$:     SEC                     ; INDICATE FAILURE
014112                                  RETURN

014114  020167  000000G         10$:    CMP     R1,$LSTAD       ; INSERT AT END OF LIST ?
014120  103773                          BLO     5$              ; IF NO - BRANCH
014122  010167  000000G                 MOV     R1,$LSTAD       ; UPDATE NEW HIGH ADRESS
014126  016701  000000G                 MOV     $LSTRM,R1       ; SET NEW PREVIOUS NODE BLOCK POINTER
014132  010067  000000G                 MOV     R0,$LSTRM       ; UPDATE NEW LAST NODE BLOCK POINTER
014136  000241                          CLC                     ; INDICATE SUCCESS
014140                                  RETURN



```

```

                        ;+
                        ; CMPSTR - COMPARE STRINGS
                        ;
                        ; INPUTS  -
                        ;               R0,R1 ADDRESS OF FILE BLOCKS TO BE MATCHED
                        ;
                        ; OUTPUTS -
                        ;               C CLEAR, MATCH
                        ;               C SET,   NO MATCH
                        ;
                        ;-

015740  121021          CMPSTR: CMPB(R0),(R1)+          ; MATCH ON FIELD SIZE
015742  001015                  BNE     15$             ; IF NOT - BRANCH
015744  005004          5$:     CLR     R4              ; INIT BYTE COUNT
015746  112004                  MOVB    (R0)+,R4        ; GET FILED SIZE
015750  001002                  BNE     10$             ; IF FILELD LENGTH - BRANCH

015752  005305                  DEC     R5              ; COUNT ONE LESS FIELD
015754  000771                  BR      CMPSTR          ; CHECK NEXT FIELD

015756  122021          10$:    CMPB    (R0)+,(R1)+     ; MATCH ?
015760  001006                  BNE     15$             ; IF NOT - BRANCH

015762  005304                  DEC     R4              ; COUNT ONE LESS CHARACTER
015764  001374                  BNE     10$             ; IF MORE - BRANCH

015766  005305                  DEC     R5              ; COUNT ONE LESS FIELD
015770  001363                  BNE     CMPSTR          ; IF MORE BRANCH
015772  000241                  CLC                     ; INDICATE BLOCK MATCH
015774  000401                  BR      20$             ; CONTINUE

015776  000261          15$:    SEC                     ; INDICATE NO MATCH
016000                  20$:    RETURN



                        ;+
                        ; INIBUF - INITIALIZE BUFFER
                        ;
                        ; INPUTS -
                        ;               R0 - BUFFER ADDRESS
                        ;               R1 - BYTE COUNT
                        ;
                        ; OUTPUTS -
                        ;               BUFFER INIT
                        ;               R0 PRESERVED
                        ;-

016002  010046          INIBUF: MOV     R0,-(SP)        ; SAVE BUFFER ADDRESS

016004  105020          10$:    CLRB    (R0)+           ; CLEAR BYTE
016006  005301                  DEC     R1              ; COUNT ONE LESS BYTE
016010  001375                  BNE     10$             ; IF MORE - BRANCH
016012  012600                  MOV     (SP)+,R0        ; RESTORE BUFFER ADDRESS

```

```
                                                .SBTTL  REQUEST NETLOADER TO LOAD A LINE
                                        ;+
                                        ; **-LDLINE-REQUEST NETLOADER TO LOAD A LINE
                                        ;
                                        ; A SYSTEM LINE TABLE ENTRY INDICATES THAT THE LINE SHOULD BE
                                        ; LOADED AT SYSTEM INITIALISATION TIME, REQUEST NETLOADER TO
                                        ; LOAD THE LINE.
                                        ;-
                                        ; INPUTS:
                                        ;       R1 - POINTER TO SYSTEM LINE TABLE
                                        ;       R2 - POINTER TO DDM PDV
                                        ;

017516  012767  000003  000000G LDLINE: MOV     #LS.LIN,RQB+LR.STS      ; SET FUNCTION CODE
017524  012767  000200  000000G         MOV     #OP$ALL,$OPTON          ; LOAD LINE AND NOTHING ELSE
017532  012767  000000G 000000G         MOV     #CM$SET,$CMAND          ; THIS IS A "SET" OPERATION
017540  005067  000000G                 CLR     $QUAL                   ; CLEAR THE QUALIFIERS
017544  005067  000000G                 CLR     TOPFLG                  ; TURN OFF TOP-DOWN LOADING
017550  005767  000000G                 TST     TOPFLG                  ; TOP-DOWN LOADING?
017554  001403                          BEQ     5$                      ; IF EQ, NO
017556  052767  002000  000000G         BIS     #LS.TOP,RQB+LR.STS
017564  116167  000012  000004G 5$:     MOVB    L.CTL(R1),RQB+LR.CTL
017572  116167  000013  000005G         MOVB    L.UNT(R1),RQB+LR.UNT
017600  105067  000006G                 CLRB    RQB+LR.TRI              ; NO TRIBUTARY
017604  016267  000004  000002G         MOV     Z.NAM(R2),RQB+LR.LIN    ; GET NAME
017612                                  CALL    REQNTL                  ; REQUEST NETLOADER TO LOAD PROCESSES
017616  005767  000000G                 TST     RQB                     ; TEST FOR ERROR
017622  100004                          BPL     10$
017624                                  NTIERR  16                      ; FAILED TO LOAD LINE
017634                          10$:    RETURN

```

```
Symbol table

UU.RDY= 000200          U.SPC = 000036          U2.PRV= 000010          $DPAS   001356RG        $RDBSZ= ****** GX
UU.SER= 000001          U.STS   000005          U2.RMT= 020000          $DUSE   001572RG        $RET    002564RG
UU.SIO= 001000          U.ST2   000007          U2.R04= 100000          $EXNOD  001650RG        $SAVVR= ****** GX
UU.SPC= 000040          U.SUB = 000036          U2.SCS= 000004          $FLAGS= ****** GX       $SCOND  003636RG
U.ACB   000062          U.TCHP  000042          U2.SLV= 000200          $GETRV= ****** GX       $SCOP1  003606RG
U.ACP = 000032          U.TCVP  000043          U2.VT5= 000002          $GRO    001706RG        $SCOP2  003616RG
U.ADMA  000066          U.TFLK  000040          U2.7CH= 010000          $GTPLN= ****** GX       $SCOP3  003626RG
U.AFLG  000064          U.TFRQ  000036          U3.OPA= 100000          $HLPAC  006276RG        $SCSR   003750RG
U.ATT   000022          U.TIXL  000060          U3.PAR= 040000          $HMBUF= ****** GX       $SEG    004026RG
U.BPKT= 000044          U.TLPP  000053          U3.UPC= 020000          $HTM    001746RG        $SLTMA= ****** GX
U.BUF   000024          U.TMTI  000047          U4.CR = 000100          $IFALI  002050RG        $SLTNM= ****** GX
U.CBF = 000032          U.TRCK= 000100          VF.ADD= ****** GX       $IFLOG  002042RG        $STADD  004102RG
U.CLI   177772          U.TSTA  000026          VF.EAD= ****** GX       $IFNNA  002004RG        $STALI  004234RG
U.CNT   000030          U.TST5  000054          VF.INI= ****** GX       $IFXAC  002056RG        $STDES  004246RG
U.COTQ  000030          U.TST6  000056          VF.SKA= ****** GX       $IFXPR  002070RG        $STEND  004260RG
U.CTCB  000026          U.TTAB  000050          VF.SS = ****** GX       $IFX5S  002102RG        $STHOS  004306RG
U.CTL   000004          U.TTYP  000046          V$$CTR= 001000          $IFX9S  002114RG        $STLND  004440RG
U.CTYP  000052          U.TUX   000024          WC.CNT= 000400          $INA    002516RG        $STLSH  004552RG
U.CW1   000010          U.UHVR= 000107          WC.EVE= 004000          $INC    002402RG        $STLSN  004516RG
U.CW2   000012          U.UIC   000044          WC.TRI= 002000          $LNAME  002166RG        $STNAD  004570RG
U.CW3   000014          U.UNFL= 000052          WC.UNT= 001000          $LTM    002230RG        $STNAM  004734RG
U.CW4   000016          U.UNIT  000006          X$$DBT= 000000          $MACT   002266RG        $STPAR  005026RG
U.CYL = 000104          U.UNSZ= 000114          ZF.MUX= ****** GX       $MDEA   002334RG        $STRPA  005072RG
U.DCB   000000          U.UNTI= 000060          Z.FLG = ****** GX       $MUL  = ****** GX       $STSNM  005166RG
U.FCDE= 000042          U.USVR= 000106          Z.NAM = ****** GX       $MXCO   002632RG        $STTPA  005230RG
U.FNUM= 000040          U.UTIL= 000036          $ADS    000322RG        $MXLI   002700RG        $TRI    005324RG
U.GRP = 000102          U.VCB = 000034          $ANT    000362RG        $NADD   003062RG        $TRIBA  005456RG
U.KCSR= 000032          U.VSER= 000120          $ARAD1  002746RG        $NNAM   004722RG        $TYPE = ****** GX
U.KCS6= 000034          U.2MED= 000070          $ARAD2  002762RG        $ONAM   003214RG        $UNT    005522RG
U.LUIC  177774          U2.AT.= 000020          $BFR  = ****** GX       $OPTON= ****** GX       $VEC    005674RG
U.MEDI= 000040          U2.CRT= 002000          $CAT5 = ****** GX       $OTYP   003260RG        $VFLAG= ****** GX
U.MLUN= 000050          U2.DH1= 100000          $CKLOG  000734RG        $OUT    002450RG        $VNCP   000000RG
U.MUP   177772          U2.DJ1= 040000          $CLBUF= ****** GX       $OVER   003350RG        $XMISS  005752RG
U.OWN   177776          U2.DZ1= 000100          $CLIOS= ****** GX       $OWN    003360RG        $XTRI   005372RG
U.RBNS= 000112          U2.ESC= 001000          $CMAND= ****** GX       $PCIR   003420RG        $XUNT   005570RG
U.RCTC= 000113          U2.HFF= 010000          $COP    001002RG        $PDT    001434RG        $$$ARG= 000003
U.RCTS= 000110          U2.HLD= 000040          $COST   001056RG        $PDVTA= ****** GX       $$$OST= 000014
U.RED   000002          U2.LOG= 000400          $CSR    001126RG        $PFLAG= ****** GX       .PCHAR= ****** GX
U.RED2  000034          U2.LWC= 000001          $CTL    001204RG        $PNAM   003476RG        .PNUMB= ****** GX
U.SCB   000020          U2.L3S= 000004          $DACC   001300RG        $PRI    003536RG        .PNUMH= ****** GX
U.SHST= 000076          U2.L8S= 010000          $DECPT= ****** GX       $QUAL = ****** GX       .PSTCN= ****** GX
U.SHUN= 000074          U2.NEC= 004000          $DEVHD= ****** GX       $RADDR= ****** GX       .PSTPT= ****** GX


. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        006562    001   (RW,I,LCL,REL,CON)
DATA    002624    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 17
Work  file writes: 21
Size of work file: 17101 Words  ( 67 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS
```

H 2

```
**FILE**ID**VNP




VV      VV   NN      NN   PPPPPPPP
VV      VV   NN      NN   PPPPPPPP
VV      VV   NN      NN   PP      PP
VV      VV   NN      NN   PP      PP
VV      VV   NNNN    NN   PP      PP
VV      VV   NNNN    NN   PP      PP
VV      VV   NN  NN  NN   PPPPPPPP
VV      VV   NN  NN  NN   PPPPPPPP
VV      VV   NN    NNNN   PP
VV      VV   NN    NNNN   PP
  VV  VV     NN      NN   PP                ....
  VV  VV     NN      NN   PP                ....
    VV       NN      NN   PP                ....
    VV       NN      NN   PP                ....




LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

I 2

VNP - VIRTUAL NETWORK COMMAND P MACRO V05.03b Saturday 29-Jun-85 02:45

```
                                                                                        ; MUST BE FOR ALL TRIBUTARIES
 001034  112767  177777  000006G           MOVB    #-1,RQB+LR.TRI                       ; START WITH TRIBUTARY ZERO
 001042  105267  000006G           40$:    INCB    RQB+LR.TRI                           ; GET NEXT TRIBUTARY
 001046  016746  000000G           50$:    MOV     RQB,-(SP)                            ; SAVE FUNCTION REQUESTED
 001052                                    CALL    $VRQB                                ; PROCESS REQUEST
 001056  012667  000000G                   MOV     (SP)+,RQB                            ; RESTORE FUNCTION CODE JUST IN CASE
                                                                                        ; NO ERROR PROCESSING
 001062  005302                            DEC     R2                                   ; ANY MORE TRIBUTARIES ?
 001064  003366                            BGT     40$                                  ; IF GT, YES
 001066  111667  000000G                   MOVB    (SP),$OPTON                          ; RESTORE LOW BYTE OF OPTIONS WORD
 001072  000706                            BR      15$                                  ; ELSE, GET THE NEXT LINE

 001074  005726                    60$:    TST     (SP)+                                ; CLEAN UP STACK
 001076  032767  000004  000000G           BIT     #QF$KNO,$QUAL                        ; WAS THIS FOR KNOWN LINES ?
 001104  001002                            BNE     70$                                  ; IF NE, YES .. LEAVE
 001106  005704                            TST     R4                                   ; ELSE, WERE ANY LINES FOUND
 001110  001401                            BEQ     102$                                 ; IF EQ, NO .. LINE NOT IN SYSTEM
 001112                            70$:    RETURN
                                   ;
                                   ; ERROR CONDITIONS
                                   ;
 001114                            102$:   ERRPT$  CNER1,$EROUT                         ; CIRCUIT NOT IN SYSTEM
```

```
                                                ;
                                                ; TEST FOR PROCESS ALREADY IN CORE
                                                ;
 002470  010167  000000G                        AIC:    MOV     R1,$PDVA        ; SAVE PDV ADDRESS
 002474  116102  000000G                                MOVB    Z.FLG(R1),R2    ; GET PROCESS TYPE FROM PDV STATUS BYTE
 002500  042702  000000C                                BIC     #^C<FL.DLC!FL.DDM>,R2 ; ...
 002504  042767  000000C 000000G                        BIC     #<FL.DLC!FL.DDM>,$FLAGS ; UPDATE NETLDR STATUS ACCORDINGLY
 002512  050267  000000G                                BIS     R2,$FLAGS       ; ...
 002516  016167  000000G 000000G                        MOV     Z.NAM(R1),$NAME ; SET PROCESS NAME (RAD50)
 002524  016102  000000G                                MOV     Z.PCB(R1),R2    ; GET PCB ADDRESS
 002530  001417                                         BEQ     10$             ; IF EQ, PROCESS IS NOT IN CORE
 002532  005067  000000G                                CLR     $BIAS           ; ASSUME UNMAPPED SYSTEM
 002536  005767  000000G                                TST     .MGMGE          ; IS THIS AN UNMAPPED SYSTEM ?
 002542  100412                                         BMI     10$             ; IF MI, YES
 002544                                                 GETRV   R2,,PLGTH       ; READ IN PCB IMAGE
 002562  016767  000000C 000000G                        MOV     $BFR+P.REL,$BIAS ; GET RELOCATION BIAS
 002570                                         10$:    RETURN


         000104'                                        .END    $VNP
```

```
     56                                          .MCALL  DCBDF$,LBLDF$,SCBDF$,UCBDF$,PCBDF$
     57                                  ;
     58                                  ; DEVICE CONTROL BLOCK OFFSETS (DCB)
     59                                  ;
     60 000000                                   DCBDF$
     61                                                  .GLOBL  D.DSP,D.NAM,D.PCB,D.UCB,D.UCBL,D.UNIT,D.VPWF
     62
     63                                  ;
     64                                  ; TASK LABEL BLOCK OFFSETS
     65                                  ;
     66 000000                                   LBLDF$
     67                                                  .GLOBL  L$BEXT,L$BFLG,L$BHRB,L$BLDZ,L$BMXZ,L$BSA,TS$NHD
     68                                                  .GLOBL  L$BBLK,L$BXFR,L$BPAR
     69
     70                                  ;
     71                                  ; STATUS CONTROL BLOCK OFFSETS (SCB)
     72                                  ;
     73 000000                                   SCBDF$
     74                                                  .GLOBL  S.CON,S.LHD
     75
     76                                  ;
     77                                  ; UNIT CONTROL BLOCK OFFSETS (UCB)
     78                                  ;
     79 000000                                   UCBDF$
     80                                                  .GLOBL  UC.PWF,U.CTL,U.DCB,U.RED,U.SCB
     81
     82                                  ;
     83                                  ; PARTITION CONTROL BLOCK OFFSETS
     84                                  ;
     85 000000                                   PCBDF$
     86
     87                                                  .IF NDF R$$MPL                                  ;[MP01]
     88                                                  .GLOBL  P.BUSY                                  ;[MP01]
     89                                                  .ENDC                                           ;[MP01]
     90
     91                                                  .GLOBL  P.BLKS,P.STAT,P.MAIN,P.OWN,P.REL
     92                                                  .GLOBL  P.SUB,P.SIZE,P.TCB,P.NAM,P.IOC,P.LNK,P.PRI
     93
     94         000001                           .END
```

```
                                                      .SBTTL NETWORK INITIALIZATION COLD START
                                             ;+
                                             ; *** - $VNTI - VNP NETWORK INITIALIZATION
                                             ;
                                             ; INPUTS:
                                             ;        NONE.
                                             ;
                                             ; OUTPUTS:
                                             ;        NETWORK IS INITIALIZED
                                             ;
                                             ;-
                                                      .MCALL  VNPBPT
000000                                       $VNTI::
000000                                                VNPBPT  VNTI
000000   010667   000000G                             MOV     SP,SPSAV                ; SAVE STACK POINTER VALUE
000004                                                CALL    $CLSAV                  ; SAVE GCL CONTEXT
                                             ;
                                             ; SEE IF COMMEXEC IS ALREADY LOADED
                                             ;
000010   032767   020000   000000G                    BIT     #FE.CEX,$FMASK          ; IS COMMEXEC LOADED ?
000016   001401                                       BEQ     3$                      ; NO
000020                                                RETURN                          ; YES, JUST RETURN
                                             ;
                                             ; DEFINE NTPOOL NAME ( AND SIZE IF MAPPED )
                                             ;
000022   005067   000000G                    3$:      CLR     SUBPAR                  ; INDICATE NO PCB YET
000026   005067   000000G                             CLR     SBPOOL                  ; INIT SERVER BASE FLAGS
000032   005067   000000G                             CLR     SBFLG                   ; ...
000036   005067   000000G                             CLR     SBPCB                   ; ...

000042                                                CALL    OPNCFG                  ; ATTEMPT TO OPEN CONFIG FILE
000046   103004                                       BCC     4$
000050                                                NTIERR  14
000060                                       4$:      CALL    NTPSCN                  ; SCAN FOR NTPOOL DEFS
000064   103004                                       BCC     5$
000066                                                NTIERR  13

000076                                       5$:
                                             ;
                                             ; FIND NETWORK POOL PARTITION
                                             ;
000076   005001                                       CLR     R1                      ; ASSUME NO PCB ADDRESS
000100   012700   000000G                             MOV     #NTPNAM,R0              ; ADDRESS OF RAD50 PARTITION NAME
000104   005710                                       TST     (R0)                    ; IS NAME PRESENT ?

                                                      .IF DF  R$$MPL                                          ;[MP01]
                                                      BNE     35$                     ; IF YES - BRANCH       ;[MP01]
                                                      .IFF
000106   001006                                       BNE     7$                      ; POOL NAME SPECIFIED

                                                      .ENDC   ; DF R$$MPL                                     ;[MP01]

000110   012767   063147   000000G                    MOV     #^RPOO,NTPNAM           ; USE DEFAULT NAME
000116   012767   047574   000002G                    MOV     #^RL..,NTPNAM+2         ; ...

                                                      .IF NDF R$$MPL                                          ;[MP01]

```

```
    847
    848                                                 .SBTTL INITIALIZE SMALL DATA BUFFER POOL
    849
    850                                         ;+
    851                                         ;       **-$SDBS        - INITIALIZE SMALL DATA BUFFER POOL
    852                                         ;
    853                                         ; INPUTS:
    854                                         ;       R4      - BIAS OF BASE OF FREE SPACE IN POOL
    855                                         ;       R5      - VIRTUAL ADDRESS OF BASE OF FREE SPACE
    856                                         ;
    857                                         ; OUTPUTS:
    858                                         ;       R4      - BIAS OF BASE OF FREE SPACE
    859                                         ;       R5      - VIRTUAL ADDRESS OF FREE SPACE
    860                                         ;       ALL OTHER REGISTERS PRESERVED
    861                                         ;-
    862
    863 002776  010346                          $SDBS:: MOV     R3,-(SP)                ; SAVE
    864 003000  010246                                  MOV     R2,-(SP)                ; REGISTERS
    865 003002  010146                                  MOV     R1,-(SP)
    866 003004  016701  000000G                         MOV     $SDBNM,R1               ; NUMBER OF SDBS TO ALLOCATE
    867 003010  001444                                  BEQ     90$                     ; IF ZERO, NO WORK TO DO
    868 003012  016703  000000G                         MOV     $SDBSZ,R3               ; NUMBER OF BYTES/SDB
    869 003016  066703  000000G                         ADD     .CRCOV,R3               ; COUNT CRC BYTES
    870 003022  005203                                  INC     R3                      ; MAKE IT EVEN
    871 003024  042703  000001                          BIC     #1,R3                   ;       ...
    872
    873 003030  066705  000000G                 10$:    ADD     $MAXOV,R5               ; SKIP MAXIMUM DLC OVERHEAD
    874 003034  010467  000000G                         MOV     R4,$RBLCK               ; MAP TO BUFFER
    875 003040  016767  000000G 000000G                 MOV     $SDBLH,$BFR             ; INITIALIZE BIAS
    876 003046  016767  000002G 000002G                 MOV     $SDBLH+2,$BFR+2         ; AND VIRTUAL ADDRESS
    877 003054  010546                                  MOV     R5,-(SP)                ; SAVE MAPPING BITS
    878 003056  042705  140000                          BIC     #140000,R5              ; CLEAR MAPPING BITS
    879 003062                                          PUTAD   R5,#4                   ; WRITE VALUE BACK TO NTPOOL
    880 003100  012605                                  MOV     (SP)+,R5                ; RESTORE REGISTER
    881 003102  010567  000002G                 40$:    MOV     R5,$SDBLH+2             ; INITIALIZE LISTHEAD VIRTUAL ADDRESS
    882 003106  010467  000000G                         MOV     R4,$SDBLH               ;  AND BIAS
    883
    884 003112                                          CALL    $GTBUF                  ; GET DOUBLE-WORD ADDRESS OF NEXT BUFFER
    885 003116  005301                                  DEC     R1                      ; LOOP TILL
    886 003120  001343                                  BNE     10$                     ;  DONE.
    887 003122  012601                          90$:    MOV     (SP)+,R1                ; RESTORE
    888 003124  012602                                  MOV     (SP)+,R2                ; REGISTERS
    889 003126  012603                                  MOV     (SP)+,R3
    890 003130                                          RETURN
```

```
                                                .SBTTL  ACTION ROUTINES FOR ESTIMATING NTPOOL SIZE


003632  016767  000000G 000074' .R1:    MOV     .PNUMB,NN               ; SAVE MAXIMUM NODE ADDRESS
003640                                  RETURN

003642  016767  000000G 000076' .R2:    MOV     .PNUMB,NA               ; SAVE MAX AREAS
003650                                  RETURN

003652  016767  000000G 000106' .R3:    MOV     .PNUMB,NBEAJ            ; SAVE BROADCAST ENDNODE ADJACENCIES
003660                                  RETURN

003662  016767  000000G 000022' .N1:    MOV     .PNUMB,NAMBUF           ; STORE NUMBER OF RDBS/SDBS
003670                                  RETURN

003672  016767  000022' 000000G .N2:    MOV     NAMBUF,.RDBNM           ; GET NUMBER OF RDBS FROM STORAGE


003700  016767  000000G 000000G         MOV     .PNUMB,.RDBSZ           ; GET RDB SIZE IN BYTES
003706                                  RETURN

003710  016767  000022' 000000G .N3:    MOV     NAMBUF,.SDBNM           ; GET NUMBER OF SDBS FROM STORAGE
003716  016767  000000G 000000G         MOV     .PNUMB,.SDBSZ           ; SIZE IN BYTES OF SDB
003724                                  RETURN


003726  016701  000000G         .N4:    MOV     .RDBNM,R1               ; GET NUMBER OF RDB'S
003732  016700  000000G                 MOV     .RDBSZ,R0               ; GET RDB SIZE
003736  066700  000000G                 ADD     .CRCOV,R0               ; ADD CRC BYTES
003742  066700  000000G                 ADD     .PNUMB,R0               ; ADD MAX DLC OVERHEAD
003746                                  CALL    $MUL                    ; DETERMINE TOTAL BYTES
003752  010067  000024'                 MOV     R0,NAMBUF+2             ; SAVE DOUBLE PRECISION ANSWER
003756  010167  000026'                 MOV     R1,NAMBUF+4             ; ...
003762  016700  000000G                 MOV     .SDBNM,R0               ; GET NUMBER OF SDB'S
003766  016701  000000G                 MOV     .SDBSZ,R1               ; GET SDB SIZE
003772  066701  000000G                 ADD     .CRCOV,R1               ; ADD CRC BYTES
003776  066701  000000G                 ADD     .PNUMB,R1               ; ADD DLC OVERHEAD
004002  005201                          INC     R1                      ; MAKE IT EVEN
004004  042701  000001                  BIC     #1,R1                   ; ...
004010                                  CALL    $MUL                    ; MULTIPLY THE TWO
004014  066700  000024'                 ADD     NAMBUF+2,R0             ; ADD PREVIOUS RESULT
004020  066701  000026'                 ADD     NAMBUF+4,R1             ;  DOUBLE PRECISION
004024  005500                          ADC     R0                      ;       ...
004026  062701  000077                  ADD     #77,R1                  ; ROUND UP TO BLOCK BOUNDARY
004032  005500                          ADC     R0                      ;       ...
004034  012702  000006                  MOV     #6,R2                   ; LOOP COUNT
004040  006200                  10$:    ASR     R0                      ; DOUBLE SHIFT RIGHT 6 PLACES
004042  006001                          ROR     R1                      ;       ...
004044  005302                          DEC     R2                      ;       ...
004046  001374                          BNE     10$                     ;       ...
004050  010167  000002G                 MOV     R1,NTPSIZ+2             ; SAVE RDB/SDB SIZE
004054                                  RETURN

004056  016767  000000G 000000G .FUDG:  MOV     .PNUMB,NTPSIZ           ; EXTRA BLOCKS NEEDED IN POOL

004064  005767  000074'                 TST     NN                      ; ENDNODE ?
004070  001461                          BEQ     10$                     ; IF YES - BRANCH
```

```
                                                .SBTTL  PROCESS TEXT STRING FROM FILE
                                        ;+
                                        ;       **-PROTXT-PROCESS TEXT STRING FROM FILE
                                        ;
                                        ;       A STRING HAS BEEN READ FROM THE CONFIGURATION FILE WHICH WE
                                        ;       MUST NOW PARSE TO CHECK WHETHER IT IS A FUNCTION THAT WE MUST
                                        ;       KNOW ABOUT AND IF SO WE MUST SET UP THE APPROPRIATE DATA
                                        ;       STRUCTURES IN THE COMMEXEC DATABASE.
                                        ;
                                        ; INPUTS:
                                        ;       CONFIG FILE MUST BE OPEN
                                        ;
                                        ; OUTPUTS:
                                        ;       R1,R2,R3,R4,R5 - CORRUPTED
                                        ;-

006256  005001                          PROTXT: CLR     R1              ; INITIALISE ACTION FLAGS - NO ABREVIATIONS
006260  016703  000000G                         MOV     $CLIOS,R3       ; IOSB ADDRESS
006264  016303  000002                          MOV     2(R3),R3        ; LENGTH OF LINE
006270  010367  000000G                         MOV     R3,LINSIZ       ; STORE LINE SIZE IN CASE OF ERROR
006274  001412                                  BEQ     10$             ; IGNORE NULL LINES
006276  016704  000000G                         MOV     $CLBUF,R4       ; POINT TO THE STRING TO PARSE
006302  012702  000032'                         MOV     #KWORD,R2       ; POINT TO THE KEYWORD TABLE
006306  012705  000466'                         MOV     #ISTATE,R5      ; SET UP THE INITIAL STATE
006312  005067  000000G                         CLR     MACTYP          ; INITIALISE MACRO TYPE FLAG
006316                                          CALL    .TPARS          ;   THEN PARSE THE LINE
006322                                  10$:    RETURN
```

```
   2239                                             .SBTTL  LOG$ST PARSE TABLES
   2240                                     ;+
   2241                                     ;       STATE TRANSITIONS FOR THE 'LOG$ST' MACRO
   2242                                     ;-
   2243 006324                                      STATE$  LOGST
   2244 006324                                      TRAN$   $NUMBR,$EXIT,L.LGST
```

```
                                                .SBTTL  ACTION ROUTINE: END OF BUF$DF SOURCE LINE
                                        ;+
                                        ;       **-.BUFCT-END OF BUF$DF SOURCE LINE DETECTED
                                        ;
                                        ;       THE CONTENTS OF $CCBNM, $RDBNM, AND $SDBNM ARE COPIED INTO $CCBCT,
                                        ;       $RDBCT, AND $SDBCT RESPECTIVELY.
                                        ;-
                                        ;
006424                                  .BUFCT:
006424  032767  000004  000000G                 BIT     #CF.DYN,$CXOPT          ; DYNAMIC CCB ALLOCATIONS?
006432  001003                                  BNE     10$                     ; BR IF YES, LEAVE CCB COUNT ALONE
006434  066767  000000G 000000G                 ADD     $RDBNM,$CCBNM           ; ADD NUMBER OF RDBS TO ALLOCATE
006442  016767  000000G 000000G 10$:            MOV     $CCBNM,$CCBCT           ; COPY # CCB'S
006450  016767  000000G 000000G                 MOV     $RDBNM,$RDBCT           ;
006456  016767  000000G 000000G                 MOV     $SDBNM,$SDBCT
006464  005267  000000G                         INC     $MAXOV                  ; MAKE MAXOV EVEN
006470  042767  000001  000000G                 BIC     #1,$MAXOV               ; MAKE MAXOV EVEN
006476                                          RETURN
```

```
  2974                                                  .SBTTL  ACTION ROUTINE: STATION FLAGS SEEN
  2975                                          ;+
  2976                                          ; *** - X.FLG - STATION FLAGS SEEN
  2977                                          ; *** - X.FLG1 - NO STATION FLAGS SEEN
  2978                                          ;
  2979                                          ; INPUTS:
  2980                                          ;       MRKFL = FLAGS BYTE
  2981                                          ;       SLTNUM = CURRENT SLT NUMBER
  2982                                          ;
  2983                                          ; OUTPUTS:
  2984                                          ;       S.FLG IN STATION TABLE IS UPDATED
  2985                                          ;
  2986                                          ;-
  2987
  2988 010244                                   X.FLG:
  2989 010244 016701  000000G                           MOV     SLTNUM,R1               ; GET THE SLT NUMBER
  2990 010250 006301                                    ASL     R1                      ; CONVERT TO A WORD INDEX
  2991 010252 066701  000000G                           ADD     $SLTMA,R1               ; POINT INTO SLT MAPPING TABLE
  2992 010256 011101                                    MOV     (R1),R1                 ; POINT TO SLT
  2993 010260 105761  000014                            TSTB    L.NSTA(R1)              ; IS THIS LINE A SLAVE LINE ?
  2994 010264 001424                                    BEQ     10$                     ; IF EQ, YES
  2995 010266 016700  000036'                           MOV     STAN,R0                 ; GET STATION NUMBER
  2996 010272 006300                                    ASL     R0                      ; CONVERT TO A DOUBLE WORD INDEX
  2997 010274 006300                                    ASL     R0                      ; ...
  2998 010276 060100                                    ADD     R1,R0                   ; GET START OF STATION TABLES
  2999 010300 062700  000022                            ADD     #L.MPF,R0               ; ...
  3000 010304 142760  000100  000000                    BICB    #SF.ENA,S.FLG(R0)       ; CLEAR SETTABLE BITS
  3001 010312 156760  000040' 000000                    BISB    MRKFL,S.FLG(R0)         ; SET FLAG BITS
  3002 010320 032761  004000  000000                    BIT     #LF.MFL,L.FLG(R1)       ; IS THE LINE MARKED FOR LOAD ?
  3003 010326 001003                                    BNE     10$                     ; IF NE, YES OKAY
  3004 010330 142760  000100  000000                    BICB    #SF.ENA,S.FLG(R0)       ; ELSE, MAKE SURE WE DON'T ENABLE LINE
  3005 010336                                  10$:
  3006 010336                                  X.FLG1: RETURN
  3007
```

```
                                                .SBTTL  ACTION ROUTINE: EVENT CLASS NUMBER FOUND

                                        ;+
                                        ;
                                        ; E.NUM - EVENT CLASS NUMBER FOUND
                                        ;
                                        ; INPUTS:
                                        ;       .PNUMB - EVENT CLASS NUMBER
                                        ;
                                        ;-
012554  005767  000000G                 E.NUM:  TST     .PNUMH          ; MUST BE WORD VALUE
012560  001022                                  BNE     10$             ; BR IF NO
012562  022767  077700  000000G                 CMP     #77700,.PNUMB   ; LEGAL VALUE?
012570  103416                                  BLO     10$             ; BR IF NO
012572  016767  000000G 000042'                 MOV     .PNUMB,CLASS    ; STORE CLASS
012600  005067  000044'                         CLR     FLTFLG          ; INITIALIZE FILTER FLAGS
012604  012700  000046'                         MOV     #EVENT,R0       ; POINT TO EVENT MASKS
012610  010067  000056'                         MOV     R0,STRADD       ; SAVE ADDRESS OF EVENT MASKS
        000004                                  .REPT   4
                                                CLR     (R0)+           ; INITIALIZE EVENT MASKS
                                                .ENDR
012624  000402                                  BR      20$             ; AND EXIT
012626  062716  000002                  10$:    ADD     #2,(SP)         ; REJECT THE TRANSITION
012632                                  20$:    RETURN
```

```

                                              .SBTTL  FIL$DF/SER$DF ACTION ROUTINES



                                       ;+
                                       ; FILES - STORE FILE NAME IN FILBUF
                                       ;
                                       ; INPUTS -
                                       ;                 STRAD - CURRENT STRING ADDRESS
                                       ;                 STRSIZ - STRING SIZE
                                       ; OUTPUTS -
                                       ;                 FILE NAME STORED IN FILBUF
                                       ;-

 014142  012702  000222'               FILES:: MOV    #FILBUF,R2        ; GET FILE NAME BUFFER ADDRESS
 014146  116722  000204'                       MOVB   STRSIZ,(R2)+      ; STORE LENGTH OF STRING
 014152  016700  000332'                       MOV    STRAD,R0          ; GET ADDRESS OF STRING

 014156  112022                        1$:     MOVB   (R0)+,(R2)+       ; COPY STRING INTO BUFFER
 014160  005367  000204'                       DEC    STRSIZ            ; COUNT ONE LESS CHARACTER
 014164  001374                                BNE    1$                ; IF MORE - BRANCH
 014166  012767  000266' 000334'               MOV    #FNBUFF,STRPT     ; RESET BUFFER ADDRESS
 014174  005067  000204'                       CLR    STRSIZ            ; INIT STRING SIZE

 014200                                        RETURN

                                       ;+
                                       ; EXT - STORE FILE EXTENSION
                                       ;
                                       ; INPUTS -
                                       ;         STRAD - STRING ADDRESS
                                       ;         STRSIZ - STRING SIZE
                                       ;
                                       ; OUTPUTS -
                                       ;         FILBUF CONTAINS FILE EXTENSION
                                       ;-

 014202  022767  000003  000204' EXT::         CMP    #EXTN,STRSIZ      ; CHECK FOR LEGAL SIZE
 014210  100432                                BMI    10$               ; IF TOO LARGE - BRANCH
 014212  016700  000332'                       MOV    STRAD,R0          ; GET STRING ADDRESS
 014216  005002                                CLR    R2                ; INIT REG FOR FILENAME SIZE
 014220  116702  000222'                       MOVB   FILBUF,R2         ; GET SIZE OF FILE NAME
 014224  012701  000222'                       MOV    #FILBUF,R1        ; GET FILE NAME BUFFER ADDRESS
 014230  005201                                INC    R1                ; SKIP SIZE BYTE
 014232  060102                                ADD    R1,R2             ; CALCULATE STARTING ADDR. OF EXTENSION
 014234  112722  000056                        MOVB   #'.,(R2)+         ; INSERT DOT
 014240  005267  000222'                       INC    FILBUF            ; BUMP BYTE COUNT
 014244  066767  000204' 000222'               ADD    STRSIZ,FILBUF     ; ADJUST FILE NAME SIZE
 014252  112022                        5$:     MOVB   (R0)+,(R2)+       ; COPY FILE EXTENSION
 014254  005367  000204'                       DEC    STRSIZ            ; COUNT ONE LESS BYTE
 014260  001374                                BNE    5$                ; IF MORE - BRANCH

 014262  012767  000266' 000334'               MOV    #FNBUFF,STRPT     ; RESET BUFFER POINTER
 014270  005067  000204'                       CLR    STRSIZ            ; INIT STRING SIZE

 014274                                        RETURN
```

```
  4347 016014                                   RETURN
```

```
   4796                                                   .SBTTL  REQUEST NETLOADER TO LOAD A DLM CIRCUIT
   4797                                          ;+
   4798                                          ;        **-LDDLM-REQUEST NETLOADER TO LOAD A DLM CIRCUIT
   4799                                          ;
   4800                                          ;        A SYSTEM LINE TABLE ENTRY INDICATES THAT THE DLM CIRCUIT SHOULD BE
   4801                                          ;        LOADED AT SYSTEM INITIALIZATION TIME.  REQUEST NETLOADER TO LOAD
   4802                                          ;        THE CIRCUIT.
   4803                                          ;-
   4804                                          ;        INPUTS:
   4805                                          ;                 R1 - POINTER TO SYSTEM LINE TABLE
   4806                                          ;                 R2 - DDM'S PDV ADDRESS
   4807                                          ;
   4808 017636  004567  000000G          LDDLM:  JSR      R5,$SAVRG                  ; SAVE R3-R5
   4809 017642  116104  000014                   MOVB     L.NSTA(R1),R4              ; GET NUMBER OF CIRCUITS
   4810 017646  005005                           CLR      R5                         ; INITIALIZE COUNT
   4811 017650  010103                           MOV      R1,R3                      ; POINT TO SLT
   4812 017652  062703  000022                   ADD      #L.MPF,R3                  ; POINT TO FIRST CIRCUIT IN SLT
   4813 017656  132763  000040  000000  10$:     BITB     #SF.MFL,S.FLG(R3)          ; LOAD THIS CIRCUIT?
   4814 017664  001447                           BEQ      30$                        ; BR IF NO
   4815 017666  012767  000033  000000G          MOV      #LS.CIR,RQB+LR.STS         ; SET FUNCTION CODE
   4816 017674  005067  000000G                  CLR      $OPTON                     ; LOAD CIRCUIT AND NOTHING ELSE
   4817 017700  012767  000000G 000000G          MOV      #CM$SET,$CMAND             ; THIS IS A ''SET'' OPERATION
   4818 017706  005067  000000G                  CLR      $QUAL                      ; CLEAR THE QUALIFIERS
   4819 017712  005067  000000G                  CLR      TOPFLG                     ; TURN OFF TOP-DOWN LOADING
   4820 017716  005767  000000G                  TST      TOPFLG                     ; TOP-DOWN LOADING?
   4821 017722  001403                           BEQ      20$                        ; IF EQ, NO
   4822 017724  052767  002000  000000G          BIS      #LS.TOP,RQB+LR.STS         ; SET TOP DOWN FLAG
   4823 017732  116167  000012  000004G 20$:     MOVB     L.CTL(R1),RQB+LR.CTL       ; STORE CONTROLLER NUMBER
   4824 017740  116167  000013  000005G          MOVB     L.UNT(R1),RQB+LR.UNT       ; STORE UNIT NUMBER
   4825 017746  110567  000006G                  MOVB     R5,RQB+LR.TRI              ; STORE STATION NUMBER
   4826 017752  016267  000004  000002G          MOV      Z.NAM(R2),RQB+LR.LIN       ; GET NAME
   4827 017760                                   CALL     REQNTL                     ; REQUEST NETLOADER TO LOAD PROCESSES
   4828 017764  005767  000000G                  TST      RQB                        ; TEST FOR ERROR
   4829 017770  100005                           BPL      30$                        ; BR IF SUCCESS
   4830 017772                                   NTIERR   31                         ; FAILED TO LOAD CIRCUIT
   4831 020002  000410                           BR       40$                        ; EXIT
   4832 020004  142763  000040  000000  30$:     BICB     #SF.MFL,S.FLG(R3)          ; CLEAR MARKED-FOR-LOAD BIT
   4833 020012  005205                           INC      R5                         ; UPDATE CIRCUIT COUNT
   4834 020014  062703  000004                   ADD      #S.LEN,R3                  ; POINT TO NEXT CIRCUIT
   4835 020020  005304                           DEC      R4                         ; MORE CIRCUITS?
   4836 020022  003315                           BGT      10$                        ; BR IF YES
   4837 020024                          40$:     RETURN
```

```
VNCP - VNP GET COMMANDS NCP STY MACRO V05.03b Saturday 29-Jun-85 02:44  Page 15-4
Symbol table

Elapsed time: 00:00:51.11
SY:VNCP.V2,[132,134]VNCP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VNCP
```

```
VNCP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 1
```

Table of contents

```
     5-   55    MACRO DEFINITIONS
     6-   78    ERROR MESSAGES
     7-   90    $VNP - VIRTUAL NCP ENTRY POINT
     8-  153    $PANIC - VNP EXIT AND PANIC EXIT ROUTINE
     9-  161    PROC - PROCESS SPECIAL REQUESTS
    10-  192    PRLIN - SPECIAL PROCESSING FOR LINES
    11-  264    $VRQB - PROCESS REQUESTS FROM VNCP
    12-  312    PRCIR - SPECIAL PROCESSING FOR CIRCUITS
    13-  392    FUNCTION CODE DISPATCH TABLE
    14-  467    NXTLIN - GET THE NEXT LINE IN THE SYSTEM
    15-  494    ECHO - ECHO COMMANDS
    16-  556    $SPRO - SET PROCESS
    17-  622    $SCIR - SET CIRCUIT
    18-  681    $SLIN - SET LINE
    19-  716    LDLIN - LOAD A LINE
```

```
                                        .SBTTL  FUNCTION CODE DISPATCH TABLE
                                ;+
                                ; DEFINED FUNCTION CODES:
                                ;
                                ;               LS.CEX=1                ; "SET CEX (PART OF LS.NTI)"
                                ;               LS.PRO=2                ; "SET PROCESS"
                                ;               LS.LIN=3                ; "SET LINE"
                                ;               LX.CEX=4                ; CLEAR CEX,PROCESSES,LINES...
                                ;               LX.PRO=5                ; "CLEAR PROCESS"
                                ;               LX.LIN=6                ; "CLEAR LINE"
                                ;               LS.LMC=7                ; RELOAD ALL KMC MICROCODE (AFTER POWERFAIL)
                                ;               LS.TST=10               ; TEST ( VNP TEST ROUTINES )
                                ;               LS.XIT=11               ; EXIT FROM VNP
                                ;               LS.HLP=12               ; HELP
                                ;               LS.NTI=13               ; "SET SYSTEM"
                                ;               LS.NOD=14               ; "SET NODE"
                                ;               LX.NOD=15               ; "CLEAR NODE"
                                ;               LS.ALI=16               ; "SET ALIAS"
                                ;               LX.ALI=17               ; "CLEAR ALIAS"
                                ;               LS.LOG=20               ; "SET LOGGING"
                                ;               LX.LOG=21               ; "CLEAR LOGGING"
                                ;               LS.OBJ=22               ; "SET OBJECT"
                                ;               LX.OBJ=23               ; "CLEAR OBJECT"
                                ;               LD.CEX=24               ; "SHOW SYSTEM"
                                ;               LD.LIN=25               ; "SHOW LINE"
                                ;               LD.PRO=26               ; "SHOW PROCESS"
                                ;               LD.NOD=27               ; "SHOW NODE"
                                ;               LD.ALI=30               ; "SHOW ALIAS"
                                ;               LD.LOG=31               ; "SHOW LOGGING"
                                ;               LD.OBJ=32               ; "SHOW OBJECT"
                                ;               LS.CIR=33               ; "SET CIRCUIT"
                                ;               LX.CIR=34               ; "CLEAR CIRCUIT"
                                ;               LD.CIR=35               ; "SHOW CIRCUIT"
                                ;
                                ;       STATUS BITS DEFINED IN HIGH BYTE
                                ;
                                ;               LS.OPT=400              ; OPTIONAL PARAMETERS INCLUDED
                                ;               LS.ECH=1000             ; ECHO THE TEMPLATE
                                ;
                                ;
                                ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                                ;-

 001124 000000G                 FTAB:   .WORD   $SCEX           ; 1 - SET CEX (PART OF LS.NTI)
 001126 001614'                         .WORD   $SPRO           ; 2 - SET PROCESS
 001130 002236'                         .WORD   $SLIN           ; 3 - SET LINE
 001132 000000G                         .WORD   $CCEX           ; 4 - CLEAR CEX
 001134 000000G                         .WORD   $CPRO           ; 5 - CLEAR PROCESS
 001136 000000G                         .WORD   $CLIN           ; 6 - CLEAR LINE
 001140 000000                          .WORD   0               ; 7 - RELOAD KMC MICROCODE AFTER POWERFAIL
 001142 000000G                         .WORD   $VTST           ; 10 - VNP TEST ROUTINES
 001144 000130'                         .WORD   $PANIC          ; 11 - EXIT FROM VNP
 001146 000000G                         .WORD   $VHELP          ; 12 - HELP
 001150 000000G                         .WORD   $VNTI           ; 13 - SET SYSTEM
 001152 000000G                         .WORD   $SNOD           ; 14 - SET NODE
 001154 000000G                         .WORD   $CNOD           ; 15 - CLEAR NODE
 001156 000000G                         .WORD   $VALI           ; 16 - SET ALIAS
```

```
                                                                 I  4
VNP - VIRTUAL NE'WORK COMMAND P MACRO V05.03b Saturday 29-Jun-85 02:45  Page 20-1
Symbol table

AIC     002470R         LO.CON  000006          LR.UIC  000020          N$$NCT= 000001          OP$SIN= 000010
A$$CHK= 000000          LO.COP  000010          LR.UNT  000005          N$$PEM= 000001          OP$STA= 000004
A$$CPS= 000000          LO.COS  000020          LS.ALI= 000016          OF$INS= 000001          OP$TPA= 000020
A$$PRI= 000000          LO.CSR  000010          LS.CEX= 000001          OF$LOG= 000100          OP$TRI= 000040
A$$TRP= 000000          LO.CTL  000032          LS.CIR= 000033          OF$OFF= 000002          OP$UCS= 000020
CIRTAB  002174R         LO.DES  000002          LS.CXO= 010000          OF$ON = 000000          OP$USE= 000002
CLEAR   001604R         LO.DTE  000047          LS.CX5= 000400          OF$SMC= 000200          OP$VEC= 000040
CL.OPN= ****** GX       LO.EVT  000004          LS.ECH= 001000          ONOFF   001566R         OP$VER= 000040
CNER1   000102R         LO.GRO  000024          LS.FDX= 004000          OP$ADD= 000200          OP$XAC= 000001
C$$CKP= 000000          LO.HOS  000040          LS.HDX= 010000          OP$ALL= 000200          OP$XPR= 000002
C$$ORE= 000400          LO.HTM  000032          LS.HLP= 000012          OP$CCS= 000004          OP$X5S= 000004
C$$RSH= 177564          LO.INA  000062          LS.LIN= 000003          OP$CIE= 004000          OP$X9S= 000010
D$$BUG= 177514          LO.INC  000060          LS.LMC= 000007          OP$CON= 000400          PRCIR   000652R
D$$ISK= 000000          LO.LDT  000046          LS.LOG= 000020          OP$COP= 000001          PRI     001612R
D$$L11= 000001          LO.LID  000030          LS.MOD= 000036          OP$COS= 000100          PRLIN   000202R
D$$YNC= 000000          LO.LIN  000004          LS.NOD= 000014          OP$DST= 000200          PROC    000134R
D$$YNM= 000000          LO.LTM  000034          LS.NTI= 000013          OP$DTE= 000040          PRTBL   000164R
ECHO    001250R         LO.MAC  000022          LS.OBJ= 000022          OP$DUP= 000002          P$$P45= 000000
ECH2    001472R         LO.MDE  000024          LS.OPT= 000400          OP$EST= 000100          P$$WRD= 000000
ECH3    001546R         LO.NAM  000044          LS.PRO= 000002          OP$EVE= 010000          P.REL = ****** GX
E$$XPR= 000000          LO.NET= 000002          LS.TOP= 002000          OP$FIL= 000200          QF$ACT= 000001
FL.DDM= ****** GX       LO.NNM  000022          LS.TST= 000010          OP$GRP= 000100          QF$ALL= 000002
FL.DLC= ****** GX       LO.OFL  000012          LS.UNF= 020000          OP$HOS= 000001          QF$KNO= 000004
FL.LLC= ****** GX       LO.ONM  000004          LS.XIT= 000011          OP$HTM= 000002          QF$LOO= 000010
FL.PX1= ****** GX       LO.OUT  000056          LX.ALI= 000017          OP$INA= 004000          Q$$OPT= 000010
FL.PX2= ****** GX       LO.OWN  000026          LX.CEX= 000004          OP$INC= 001000          RQB   = ****** GX
FTAB    001124R         LO.PAR  000010          LX.CIR= 000034          OP$KDS= 002000          R$$DER= 000000
F$$LVL= 000001          LO.PRI  000012          LX.LIN= 000006          OP$KDT= 000400          R$$K11= 000001
F.RSIZ= ****** GX       LO.RET  000064          LX.LOG= 000021          OP$KEV= 001000          R$$SND= 000000
G$$TPP= 000000          LO.RPA  000030          LX.NOD= 000015          OP$KGR= 001000          R$$11M= 000000
G$$TSS= 000000          LO.SCR  000016          LX.OBJ= 000023          OP$KNT= 010000          SC      001572R
G$$TTK= 000000          LO.SEG  000054          LX.PRO= 000005          OP$KSK= 002000          SET     001600R
G$$WRD= 000000          LO.SKA  000014          L$$ASG= 000000          OP$LCT= 000200          SYSFDB= ****** GX
I$$RAR= 000000          LO.SKN  000014          L$$DRV= 000000          OP$LNE= 000040          S$$WRG= 000000
I$$RDN= 000000          LO.SNM  000035          L$$P11= 000001          OP$LOC= 000001          S$$YSZ= 007600
K$$CNT= 177546          LO.TPA  000020          L$$11R= 000000          OP$LST= 000002          TY$CHA= 000400
K$$CSR= 177546          LO.TRB  000034          L.CTL = ****** GX       OP$LTM= 000001          TY$EVE= 001000
K$$LDC= 000000          LO.TRI  000030          L.DDM = ****** GX       OP$LVL= 000010          TY$STA= 002000
K$$TPS= 000074          LO.UNT  000033          L.FLG = ****** GX       OP$MAC= 000010          TY$SUM= 000000
LDLIN   002320R         LO.VCT  000014          L.NSTA= ****** GX       OP$MCO= 000002          T$$KMG= 000000
LD$LP = 000000          LO.VER  000042          L.UNT = ****** GX       OP$MDE= 000400          T$$MIN= 000000
LD.ALI= 000030          LR.ACC  000020          MS.CIR= ****** GX       OP$MLI= 000004          UNS1    000650R
LD.CEX= 000024          LR.ACO  000052          MS.EXT= ****** GX       OP$MON= 000100          VF.ECH= ****** GX
LD.CIR= 000035          LR.ADD  000012          MS.INC= ****** GX       OP$MST= 000020          VF.NTI= ****** GX
LD.LIN= 000025          LR.ALI  000002          M$$CRB= 000124          OP$NAM= 000004          V$$CTR= 001000
LD.LOG= 000031          LR.BLK  000010          M$$CRX= 000000          OP$NET= 004000          WC.CNT= 000400
LD.MOD= 000037          LR.CTL  000004          M$$FCS= 000000          OP$NLI= 000002          WC.EVE= 004000
LD.NOD= 000027          LR.DES  000010          M$$MGE= 000000          OP$NOD= 000020          WC.TRI= 002000
LD.OBJ= 000032          LR.HLP  000002          M$$NET= 000000          OP$OUT= 002000          WC.UNT= 001000
LD.PRO= 000026          LR.LIN  000002          M$$OVR= 000000          OP$OWN= 000020          X$$DBT= 000000
LF.RDY= ****** GX       LR.NAM  000012          NXTLIN  001222R         OP$PAR= 000010          ZF.MFL= ****** GX
LINTAB  002312R         LR.PAS  000041          N$$ACC= 000001          OP$PRI= 000010          ZF.PSE= ****** GX
LNER1   000024R         LR.PRO  000002          N$$BUF= 000001          OP$RET= 010000          Z.FLG = ****** GX
LNER2   000052R         LR.STS  000000          N$$LDV= 000001          OP$RPA= 000010          Z.NAM = ****** GX
LO.ADD  000022          LR.TRI  000006          N$$MCP= 000001          OP$SCO= 000001          Z.PCB = ****** GX
LO.ADR  000052          LR.TYP  000002          N$$MLL= 000001          OP$SEG= 000400          $ALADJ= ****** GX
LO.CLS  000002          LR.UCB  000016          N$$MOV= 000010          OP$SER= 000400          $ALECL= ****** GX
```

```
                                                                 J  4
VNP - VIRTUAL NETWORK COMMAND P MACRO V05.03b Saturday 29-Jun-85 02:45  Page 20-2
```

```
                                                                        I 5
VNSD - VNP SYMBOL DEFINITIONS   MACRO V05.03b Saturday 29-Jun-85 02:46  Page 5-1
Symbol table

AS.DEL= 000010      G$$TPP= 000000      M.UMRA  000002      R$$K11= 000001      S3.CTC= 000004
AS.EXT= 000004      G$$TSS= 000000      M.UMRN  000004      R$$SND= 000000      S3.FDX= 000200
AS.RED= 000001      G$$TTK= 000000      M.UMVH  000010      R$$11M= 000000      S3.ICE= 001000
AS.WRT= 000002      G$$WRD= 000000      M.UMVL  000006      SPARE = 000010      S3.MHE= 000400
A$$CHK= 000000      I$$RAR= 000000      N$$ACC= 000001      SP.EIP= 000001      S3.NEC= 000020
A$$CPS= 000000      I$$RDN= 000000      N$$BUF= 000001      SP.ENB= 000002      S3.PPT= 020000
A$$PRI= 000000      K$$CNT= 177546      N$$LDV= 000001      SP.LOG= 000004      S3.PTH= 004000
A$$TRP= 000000      K$$CSR= 177546      N$$MCP= 000001      S$$WRG= 000000      S3.RAL= 000010
A.IOC   000003      K$$LDC= 000000      N$$MLL= 000001      S$$YSZ= 007600      S3.RES= 010000
A.LGTH= 000014      K$$TPS= 000074      N$$MOV= 000010      S.BMSK  177776      S3.RUB= 040000
A.MPCT  000011      LB$REV= 000400      N$$NCT= 000001      S.BMSV  177774      S3.TAB= 000002
A.PCB   000012      LD$ACC= 100000      N$$PEM= 000001      S.CON   000010 G    S3.TME= 002000
A.PCBL  000000      LD$CLS= 020000      PS.APR= 000007      S.CSR   000012      S3.TSY= 000040
A.PRI   000002      LD$LP = 000000      PS.CHK= 010000      S.CTM   000006      S3.8BC= 000100
A.STAT  000010      LD$REL= 000004      PS.CKP= 040000      S.DMCS  000020      S4.ABD= 100000
A.TCB   000004      LD$RSV= 040000      PS.CKR= 020000      S.FRK   000016      S4.ANI= 000400
A.TCBL  000006      LD$SUP= 000010      PS.COM= 000200      S.ITM   000007      S4.AVO= 001000
C$$CKP= 000000      L$BASG  001000      PS.DEL= 000010      S.LHD   000000 G    S4.BLK= 002000
C$$DRE= 000400      L$BBLK  000360 G    PS.DRV= 000020      S.PKT   000014      S4.DEC= 004000
C$$RSH= 177564      L$BDAT  000032      PS.FXD= 004000      S.PRI   000004      S4.DLO= 000100
DV.CCL= 000002      L$BEXT  000352 G    PS.LIO= 001000      S.RCNT  177772      S4.EDT= 010000
DV.COM= 020000      L$BFLG  000030 G    PS.NSF= 000400      S.ROFF  177773      S4.HFF= 000020
DV.DIR= 000010      L$BFL2  000774      PS.OUT= 100000      S.STS   000011      S4.HFL= 000007
DV.EXT= 000400      L$BHGV  000012      PS.PER= 002000      S.VCT   000005      S4.HHT= 000040
DV.F11= 040000      L$BHRB  000356 G    PS.PIC= 000100      S1.DEC= 001000      S4.HSY= 000200
DV.ISP= 002000      L$BLDZ  000016 G    PS.SYS= 000040      S1.DPR= 000400      S4.RGS= 020000
DV.MBC= 000400      L$BLIB  000040      P$$P45= 000000      S1.DSI= 004000      S4.SFC= 040000
DV.MNT= 100000      L$BLRL  000776      P$$WRD= 000000      S1.ESC= 000002      S4.VFL= 000010
DV.MSD= 000100      L$BLUN  000362      P.BLKS  000016 G    S1.IBF= 002000      S5.ABP= 100000
DV.OSP= 004000      L$BMXV  000014      P.BUSY  000024 G    S1.IBY= 000200      S5.BCC= 020000
DV.PSE= 010000      L$BMXZ  000020 G    P.IOC   000003 G    S1.OBY= 000100      S5.DAO= 040000
DV.REC= 000001      L$BOFF  000022      P.LNK   000000 G    S1.PTH= 000010      S5.HPC= 000014
DV.SDI= 000020      L$BPAR  000004 G    P.MAIN  000012 G    S1.RES= 010000      S5.HPO= 000020
DV.SQD= 000040      L$BPRI  000346      P.NAM   000004 G    S1.RNE= 000020      S5.ITI= 000100
DV.SWL= 001000      L$BROB  000364      P.OWN   000026 G    S1.RNF= 020000      S5.OXF= 000040
DV.TTY= 000004      L$BROL  000366      P.PRI   000002 G    S1.RSP= 000004      S5.RPO= 002000
DV.UMD= 000200      L$BSA   000010 G    P.REL   000014 G    S1.RST= 000001      S5.SW1= 000001
D$$BUG= 177514      L$BSEG  000026      P.SIZE  000016 G    S1.TNE= 040000      S5.TMM= 000002
D$$ISK= 000000      L$BSGL  000354      P.STAT  000030 G    S1.TSY= 000040      S5.VER= 010000
D$$L11= 000001      L$BSYS  000025      P.SUB   000010 G    S1.USI= 100000      S5.XOF= 000004
D$$YNC= 000000      L$BTSK  000000      P.SWSZ  000022      S2.BEL= 020000      S5.XON= 000010
D$$YNM= 000000      L$BWND  000024      P.TCB   000026 G    S2.BRQ= 000020      S6.EIO= 000400
D.DSP   000012 G    L$BXFR  000350 G    P.WAIT  000020      S2.CR = 002000      S6.RDI= 100000
D.LNK   000000      L$$ASG= 000000      Q$$OPT= 000010      S2.CTO= 040000      S6.RLU= 001000
D.MSK   000014      L$$DRV= 000000      R$LDAT  000026      S2.CTS= 100000      TS$ACP= 020000
D.NAM   000004 G    L$$P11= 000001      R$LFLG  000024      S2.ELF= 001000      TS$CHK= 000100
D.PCB   000034 G    L$$11R= 000000      R$LHGV  000006      S2.FLF= 000400      TS$CMP= 000200
D.UCB   000002 G    M$$CRB= 000124      R$LLDZ  000012      S2.IRQ= 000200      TS$IOP= 000020
D.UCBL  000010 G    M$$CRX= 000000      R$LMXV  000010      S2.OBF= 004000      TS$NEW= 000001
D.UNIT  000006 G    M$$FCS= 000000      R$LMXZ  000014      S2.ORQ= 000100      TS$NHD= 040000 G
D.VCAN= 000002      M$$MGE= 000000      R$LNAM  000000      S2.PCU= 010000      TS$NSD= 002000
D.VDEB= 177776      M$$NET= 000000      R$LOFF  000016      S2.RCU= 000001      TS$NXH= 000002
D.VINI= 000000      M$$OVR= 000000      R$LSA   000004      S2.SRQ= 000040      TS$PIC= 100000
D.VOUT= 000004      M.BFVH  000011      R$LSEG  000022      S2.WAL= 000010      TS$PMD= 010000
D.VPWF= 000006 G    M.BFVL  000012      R$LSIZ  000034      S2.WRA= 000006      TS$PRV= 000400
E$$XPR= 000000      M.LGTH= 000014      R$LWND  000020      S2.WRB= 000002      TS$RES= 000040
F$$LVL= 000001      M.LNK   000000      R$$DER= 000000      S3.ACR= 000001      TS$SLV= 004000
```

```
                                                                        J 5
VNSD - VNP SYMBOL DEFINITIONS   MACRO V05.03b Saturday 29-Jun-85 02:46  Page 5-2
```

```
                                                ;
                                                ; FIND SPECIFIED PARTITION
                                                ;
000124  016767  000000G 000000G 7$:     MOV     $PARHD,$BFR             ; ADDRESS OF PARTITION LISTHEAD
000132  016767  000000G 000000G         MOV     .PLGTH,$RSIZE           ; SET SIZE OF TRANSFERS
000140  016767  000000G 000000G 10$:    MOV     $BFR,$RADDR             ; SET ADDRESS OF NEXT PCB
000146  001504                          BEQ     30$                     ; IF EQ, NO MORE PARTITIONS.
000150                                  GETRV                           ; GET PCB INTO DEFAULT BUFFER
000154  026710  000004G                 CMP     P.NAM+$BFR,(R0)         ; FIRST HALF MATCH ?
000160  001367                          BNE     10$                     ; NOPE
000162  026760  000006G 000002          CMP     P.NAM+2+$BFR,2(R0)      ; SECOND HALF MATCH ?
000170  001363                          BNE     10$                     ; NOPE
000172  032767  000200  000030G         BIT     #PS.COM,P.STAT+$BFR     ; IS THIS A COMMON PARTITION ?
000200  001757                          BEQ     10$                     ; NO ???

000202  105767  000025G                 TSTB    P.BUSY+1+$BFR           ; PARTITION BUSY ?
000206  001064                          BNE     30$                     ; YES
000210  016702  000000G                 MOV     $RADDR,R2               ; SAVE PCB ADDRESS
000214  016703  000012G                 MOV     P.MAIN+$BFR,R3          ; GET MAIN PCB ADDRESS
000220  020203                          CMP     R2,R3                   ; ARE THEY THE SAME?
000222  001434                          BEQ     20$                     ; IF EQ, YES - NO MAIN PCB WORK TO DO
000224  012704  000000G                 MOV     #$BFR,R4                ; POINT JUST PAST THE PCB
000230  066704  000000G                 ADD     .PLGTH,R4               ;       ...
000234                                  SWTCHI  R4                      ; READ IN THE MAIN PCB
000240                                  GETRV   R3                      ;       ...
000250                                  SWTCHI                          ;       ...
000256  105764  000025                  TSTB    P.BUSY+1(R4)            ; IS MAIN PCB BUSY?
000262  001036                          BNE     30$                     ; IF NE, YES
000264  156764  000024G 000025          BISB    P.BUSY+$BFR,P.BUSY+1(R4) ; SET MAIN PCB BUSY
000272                                  SWTCHO  R4                      ; RE-WRITE MAIN PCB
000276                                  PUTRC                           ;       ...
000302                                  SWTCHO                          ;       ...
000310  010267  000000G                 MOV     R2,$RADDR               ; RESTORE PCB ADDRESS

000314  156767  000024G 000025G 20$:    BISB    P.BUSY+$BFR,P.BUSY+1+$BFR ; SET IT BUSY
000322                                  PUTRC                           ; REWRITE PCB
000326  012702  000000G                 MOV     #$BFR,R2                ; GET BUFFER ADDRESS
000332  010203                          MOV     R2,R3                   ; COPY IT
000334  062702  000014                  ADD     #P.REL,R2               ; POINT AT RELOCATION BIAS
000340  011267  000000G                 MOV     (R2),PSTART             ; COPY POOL STARTING BLOCK NUMBER
000344  062703  000016                  ADD     #P.SIZE,R3              ; POINT AT P.BLKS/P.SIZE FIELD
000350  011367  000000G                 MOV     (R3),PBLOX              ; COPY POOL SIZE IN BLOCKS
000354  016701  000000G                 MOV     $RADDR,R1               ; LOAD ADDRESS OF POOL PARTITION
                                        ;
                                        ; IF R1 EQ 0 -  PARTITION NAME DOES NOT EXIST
                                        ; IF R1 NE 0 -  PARTITION CONTROL BLOCK ADDRESS
                                        ;
000360  005701                          30$:    TST     R1              ; GET PCB ADDRESS ?
000362  001002                                  BNE     110$            ; YES

                                                .ENDC
                                        ;
                                        ; CREATE A NEW SUBPARTITION
                                        ;
000364                                  35$:    CALL    $CRPAR          ; DYNAMICALLY CREATE PARTITION
                                        ;
                                        ; COMMON PATH
```

```
                                                .SBTTL RETURN DOUBLE-WORD ADDRESS OF NEXT FREE BUFFER

                                        ;+
                                        ;       **-$GTBUF       - RETURN ADDRESS OF NEXT BUFFER IN POOL
                                        ;
                                        ; INPUTS:
                                        ;       R3      - NUMBER OF BYTES NEEDED ( EVEN )
                                        ;       R4      - OLD BIAS
                                        ;       R5      - OLD VIRTUAL ADDRESS
                                        ;
                                        ; OUTPUTS:
                                        ;       R4      - NEW BIAS
                                        ;       R5      - NEW VIRTUAL ADDRESS
                                        ;       OTHER REGISTERS UNCHANGED
                                        ;
                                        ;-

003132                                  $GTBUF::
003132  010046                                  MOV     R0,-(SP)                ; SAVE R0
003134  060305                                  ADD     R3,R5                   ; ADD BUFFER SIZE TO VIRTUAL ADDRESS
003136  010500                                  MOV     R5,R0                   ; SAVE NUMBER
003140  042705  177700                          BIC     #^C<77>,R5              ; GET NEW VIRTUAL ADDRESS
003144  052705  140000                          BIS     #.BASEB,R5              ; FORCE PAR3 BIAS IN CASE BUF > 4K
003150  042700  160000                          BIC     #160000,R0              ; CLEAR MAPPING BIAS BITS
003154  006200                                  ASR     R0                      ; # OF FULL 32-WORD BLOCKS IN BUFFER
003156  006200                                  ASR     R0                      ; ...
003160  006200                                  ASR     R0                      ; ...
003162  006200                                  ASR     R0                      ; ...
003164  006200                                  ASR     R0                      ; ...
003166  006200                                  ASR     R0                      ; ...
003170  000241                                  CLC                             ; ...
003172  060004                                  ADD     R0,R4                   ; NEW BIAS
003174  012600                                  MOV     (SP)+,R0                ; RESTORE R0
003176                                          RETURN
```

```

004072  016700  000104'                MOV     RAD,R0                  ; STORE BROADCAST ROUTER ADJACENCIES
004076  005200                         INC     R0                      ; INCLUDE LOOPBACK CHANNEL
004100  066700  000066'                ADD     XPTLLN,R0               ; INCLUDE NUMBER OF CIRCUITS
004104  066700  000106'                ADD     NBEAJ,R0                ; INCLUDE BROADCAST ENDNODE ADJACENCIES
004110  012701  000010                 MOV     #<2*A$LEN>,R1           ; STORE LENGTH PER ADJACENCY
004114                                 CALL    $MUL                    ; CLACULATE TOTAL SIZE
004120  062701  000176                 ADD     #176,R1                 ; ROUND UP TWO BLOCKS
004124  042701  000077                 BIC     #77,R1                  ; ...
004130  000241                         CLC
        000006                         .REPT   6
                                       ASR     R1                      ; CONVERT TO BLOCKS
                                       .ENDM
004146  060167  000000G                ADD     R1,NONMAP               ; ADJUST NON UMR POOL.. SIZE
004152  060167  000000G                ADD     R1,NTPSIZ               ; ADJUST TOTAL POOL SIZE

004156  016700  000074'                MOV     NN,R0                   ; COPY MAX NODES
004162  066700  000076'                ADD     NA,R0                   ; GET TOTAL ROL ENTRIES
004166  006300                         ASL     R0                      ; CONVERT TO BYTES
004170  062700  000010                 ADD     #10,R0                  ; ALLOW FOR ZERO OFFSET
004174  062700  000077                 ADD     #77,R0                  ; ROUND UP TO NEXT BLOCK
004200  042700  000077                 BIC     #77,R0                  ; ...
004204  006300                         ASL     R0                      ; ALLOCATE MIN HOP/COST MATRIX
004206  000241                         CLC                             ; CLEAR C-BIT

        000006                         .REPT   6
                                       ASR     R0                      ; CALCULATE NUMBER OF BLOCKS
                                       .ENDM

004224  060067  000000G                ADD     R0,NONMAP               ; ALLOCATE SOME NON UMR MAPPED POOL..
004230  060067  000000G                ADD     R0,NTPSIZ               ; UPDATE POOL SIZE


004234  016700  000066'        10$:    MOV     XPTLLN,R0               ; COPY NUMBER OF CIRCUITS
004240  012701  000046                 MOV     #T$LLEN,R1              ; STORE CONTER BLOCK SIZE
004244                                 CALL    $MUL                    ; CALCULATE TOTAL
004250  062701  000077                 ADD     #77,R1                  ; ROUND UP TO NEXT BLOCK
004254  042701  000077                 BIC     #77,R1                  ; ...

        000006                         .REPT   6
                                       ASR     R1                      ; CONVERT TO BLOCKS
                                       .ENDM

004274  060167  000000G                ADD     R1,NONMAP               ; ALLOCATE NON UMR BLOCK POOL
004300  060167  000000G                ADD     R1,NTPSIZ               ; UPDATE TOTAL POOL SIZE

004304  016700  000070'                MOV     ECLLLN,R0               ; COPY MAX LOGICAL LINKS
004310  012701  000124                 MOV     #L.LNG,R1               ; STORE LOGICAL LINK SIZE
004314                                 CALL    $MUL                    ; CALCULATE TOTAL
004320  062701  000077                 ADD     #77,R1                  ; ROUND UP TO NEXT BLOCK
004324  042701  000077                 BIC     #77,R1                  ; ...

        000006                         .REPT   6
                                       ASR     R1                      ; CONVERT TO BLOCKS
                                       .ENDM

004344  010167  000000G                MOV     R1,.ALLOC               ; ALLOCATE BLOCKS OF BYTE POOL..
```

```
   1878                                              .SBTTL  STATE TRANSITION TABLES FOR PARSING
   1879                                      ;+
   1880                                      ;       THE FOLLOWING STATE TRANSITION TABLES DEFINE THE SYNTAX FOR
   1881                                      ;       PARSING THE CONFIGURATION FILE.
   1882                                      ;-
   1883
   1884 006324                                       ISTAT$  ISTATE,KWORD
   1885
   1886                                      ;+
   1887                                      ;       IN THE INITIAL STATE WE IGNORE LINES BEGINNING WITH ';'
   1888                                      ;-
   1889
   1890 006324                                       STATE$
   1891 006324                                        TRAN$   <';>,$EXIT
   1892 006324                                        TRAN$   %BUF$DF%,BUFDF,.BUFDF
   1893 006324                                        TRAN$   %LLC$DF%,PRODF,,MC.LLC,MACTYP
   1894 006324                                        TRAN$   %DLC$DF%,PRODF,,MC.DLC,MACTYP
   1895 006324                                        TRAN$   %DDM$DF%,PRODF,,MC.DDM,MACTYP
   1896 006324                                        TRAN$   %STA$DF%,STADF,.STADF,MC.STA,MACTYP
   1897 006324                                        TRAN$   %SLT$DF%,SLTDF,.SLTDF,MC.SLT,MACTYP
   1898 006324                                        TRAN$   %ADD$DF%,ADDDF,,MC.ADD,MACTYP
   1899 006324                                        TRAN$   %NOD$DF%,NODDF,,MC.NOD,MACTYP
   1900 006324                                        TRAN$   %OBJ$DF%,OBJDF,,MC.OBJ,MACTYP
   1901 006324                                        TRAN$   %EVT$DF%,EVTDF,,MC.EVT,MACTYP
   1902 006324                                        TRAN$   %LOG$ST%,LOGST,,MC.LOG,MACTYP
   1903 006324                                        TRAN$   %REM$DF%,REMDF,,MC.REM,MACTYP
   1904 006324                                        TRAN$   %ROU$DF%,ROUDF,,MC.ROU,MACTYP
   1905 006324                                        TRAN$   %SER$DF%,SERDF
   1906 006324                                        TRAN$   %FIL$DF%,FILDF
   1907 006324                                        TRAN$   %DEC$DF%,DECDF
   1908 006324                                        TRAN$   $LAMDA,$EXIT
   1909 006324                                       STATE$
```

```
                                        .SBTTL  ROU$DF PARSE TABLES
                                ;+
                                ;       STATE TRANSITIONS FOR THE 'ROU$DF' MACRO
                                ;
                                ;-

006324                                  STATE$  ROUDF
006324                                   TRAN$  $NUMBR,,.RADDR
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.RCOST
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.RHOP
006324                                  STATE$
006324                                   TRAN$   <',>
006324                                  STATE$
006324                                   TRAN$   $NUMBR,,.RRTM
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.RMAXA
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.RBEA
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.RAMXH
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,,.RAMXC
006324                                  STATE$
006324                                   TRAN$  <',>
006324                                  STATE$
006324                                   TRAN$  $NUMBR,$EXIT,.RBTM
006324                                  STATE$


```

```
   2614                                                     .SBTTL  ACTION ROUTINE: PROCESS NAME FOUND
   2615                                             ;+
   2616                                             ;       **-.PRONM-PROCESS NAME FOUND
   2617                                             ;
   2618                                             ;       A STATE TRANSITION HAS JUST PARSED A PROCESS NAME, CONVERT IT
   2619                                             ;       TO A 16-BIT RAD50 VALUE.
   2620                                             ;-
   2621                                             ;
   2622 006500  016700  000000G                     .PRONM: MOV     .PSTPT,R0       ; POINT TO THE PROCESS NAME
   2623 006504  010701                                      MOV     PC,R1           ; ALLOW '.' AS A RAD50 CHARACTER
   2624 006506                                              CALL    $CAT5           ; CONVERT NAME TO RAD50
   2625 006512  010167  000000G                             MOV     R1,PDVNAM       ; SAVE PROCESS NAME FOR LATER
   2626 006516                                              RETURN
```

```



                                        ;+
                                        ; STORE MAX LOGICAL LINKS AND NODE COUNTER VALUES
                                        ;-

010340  105767  000001G                 ECLLNK:TSTB     .PNUMB+1            ; > 255. ?
010344  001021                                  BNE     101$                ; IF YES - BRANCH
010346                                          GETRV   $DECPT,#D$END       ; GET DECNET HOME BLOCK ADDRESS
010366  156767  000000G 000040G                 BISB    .PNUMB,D$MLL+$BFR ; STORE MAX LINKS IN HOME BLOCK
010374  016767  000000G 000000G                 MOV     .PNUMB,$$LINX       ; ...
010402                                          PUTRC                       ; UPDATE HOME BLOCK
010406                                          RETURN

010410                                  101$:   NTIERR  34                  ; ILLEGAL MAX LOGICA LINKS VALUE

010420  105767  000001G                 NODCT:  TSTB    .PNUMB+1            ; > 255. ?
010424  001021                                  BNE     101$                ; IF YES - BRANCH
010426                                          GETRV   $DECPT,#D$END       ; GET DECNET HOME BLOCK ADDRESS
010446  156767  000000G 000041G                 BISB    .PNUMB,D$MNOD+$BFR ; STORE MAX NODE COUNTERS IN HOME BLOCK
010454  016767  000000G 000000G                 MOV     .PNUMB,$$NODC       ; ...
010462                                          PUTRC                       ; UPDATE HOME BLOCK
010466                                          RETURN

010470                                  101$:   NTIERR  35                  ; ILLEGAL MAX NODE COUNTERS VALUE

010500  105767  000001G                 INCT:   TSTB    .PNUMB+1            ; > 255. ?
010504  001016                                  BNE     101$                ; IF YES - BRANCH
010506                                          GETRV   $DECPT,#D$END       ; GET DECNET HOME BLOCK ADDRESS
010526  156767  000000G 000042G                 BISB    .PNUMB,D$INCT+$BFR ; STORE INCOMING TIMER VALUE
010534                                          PUTRC                       ; UPDATE HOME BLOCK
010540                                          RETURN

010542                                  101$:   NTIERR  42                  ; ILLEGAL INCOMING TIMER VALUE

010552  105767  000001G                 OUTT:   TSTB    .PNUMB+1            ; > 255. ?
010556  001016                                  BNE     101$                ; IF YES - BRANCH
010560                                          GETRV   $DECPT,#D$END       ; GET DECNET HOME BLOCK ADDRESS
010600  156767  000000G 000043G                 BISB    .PNUMB,D$OUTT+$BFR ; STORE OUTGOING TIMER VALUE
010606                                          PUTRC                       ; UPDATE HOME BLOCK
010612                                          RETURN


010614                                  101$:   NTIERR  43                  ; ILLEGAL OUTGOING TIMER

010624  105767  000001G                 INAC:   TSTB    .PNUMB+1            ; > 255. ?
010630  001016                                  BNE     101$                ; IF YES - BRANCH
010632                                          GETRV   $DECPT,#D$END       ; GET HOME BLOCK ADDRESS
010652  156767  000000G 000044G                 BISB    .PNUMB,D$INAC+$BFR ; STORE INACTIVITY TIMER
010660                                          PUTRC                       ; UPDATE HOME BLOCK
010664                                          RETURN

010666                                  101$:   NTIERR  44                  ; ILLEGAL INACTIVITY TIMER

010676  105767  000001G                 DELF:   TSTB    .PNUMB+1            ; > 255. ?
010702  001016                                  BNE     101$                ; IF YES - BRANCH
010704                                          GETRV   $DECPT,#D$END       ; GET HOME BLOCK ADDRESS
```

```
   3453                                                 .SBTTL  ACTION ROUTINE: EVENT TYPE MASK SEEN
   3454
   3455                                       ;+
   3456                                       ;
   3457                                       ; E.MSK - EVENT TYPE MASK SEEN
   3458                                       ;
   3459                                       ; INPUTS:
   3460                                       ;         .PNUMB - EVENT TYPE MASK
   3461                                       ;
   3462                                       ;-
   3463 012634  005767  000000G               E.MSK:  TST     .PNUMH                  ; MUST BE WORD VALUE
   3464 012640  001013                                BNE     10$                     ; BR IF NO
   3465 012642  022767  000056' 000056'               CMP     #EVENT+10,STRADD        ; MORE THAN 4 EVENT TYPE MASK WORDS?
   3466 012650  001407                                BEQ     10$                     ; BR IF YES - ERROR
   3467 012652  016777  000000G 000056'               MOV     .PNUMB,@STRADD          ; SAVE THE MASK
   3468 012660  062767  000002  000056'               ADD     #2,STRADD               ; POINT TO NEXT EVENT TYPE MASK
   3469 012666  000402                                BR      20$                     ; AND EXIT
   3470 012670  062716  000002                10$:    ADD     #2,(SP)                 ; REJECT THE TRANSITION
   3471 012674                                20$:    RETURN
   3861
   3862 014276  062716  000002           10$:    ADD     #2,(SP)          ; SYNTAX ERROR
   3863 014302                                    RETURN
   3864
   3865
```

```
                                   ;+
                                   ; MOVBUF - SET UP LOCAL FILE NAME BLOCK
                                   ;
                                   ; INPUTS -
                                   ;                R2 - INPUT BUFFER
                                   ;                R0 - LOCAL FILE NAME BLOCK ADDRESS
                                   ;
                                   ; OUTPUTS -
                                   ;                LOCAL FILE BLOCK SET
                                   ;-

 016016  005701                    MOVBUF: TST     R1              ; ZERO LENGTH ?
 016020  001002                            BNE     10$             ; IF NOT - BRANCH
 016022  111220                            MOVB    (R2),(R0)+      ; COPY THE ZERO LENGTH BYTE (DLL DEFAULT)
 016024  000404                            BR      30$             ; CONTINUE

 016026  005201                    10$:    INC     R1              ; INCLUDE STRING LENGTH BYTE
 016030  112220                    20$:    MOVB    (R2)+,(R0)+     ; COPY FIELD
 016032  005301                            DEC     R1              ; ANY LEFT ?
 016034  001375                            BNE     20$             ; IF MORE - BRANCH

 016036                            30$:    RETURN

```

```
                                                   .SBTTL  REQUEST NETLOADER TO LOAD A PROCESS
                                            ;+
                                            ;      **-LDPRO-REQUEST NETLOADER TO LOAD A PROCESS
                                            ;
                                            ;      A PDV HAS BEEN FOUND TO HAVE THE 'MARKED FOR LOAD' BIT SET.
                                            ;      REQUEST NETLOADER TO LOAD THE PROCESS.
                                            ;-
                                            ;      INPUTS:
                                            ;              R2 - PDV POINTER
                                            ;
020026  012767  000002  000000G LDPRO:      MOV     #LS.PRO,RQB+LR.STS      ; SET FUNCTION CODE
020034  005067  000000G                     CLR     $OPTION                 ; NO OPTIONS
020040  012767  000000G 000000G             MOV     #CM$SET,$CMAND          ; THIS IS A "SET" OPERATION
020046  005067  000000G                     CLR     $QUAL                   ; CLEAR ALL QUALIFIERS
020052  005067  000000G                     CLR     TOPFLG                  ; TURN OFF TOP-DOWN LOADING
020056  005767  000000G                     TST     TOPFLG                  ; TOP-DOWN LOADING?
020062  001403                              BEQ     5$                      ; IF EQ, NO
020064  052767  002000  000000G             BIS     #LS.TOP,RQB+LR.STS
020072  016267  000004  000002G 5$:         MOV     Z.NAM(R2),RQB+LR.PRO
020100  005067  000004G                     CLR     RQB+LO.LIN              ; INITIALIZE NTL REQUEST BLOCK
020104  005067  000006G                     CLR     RQB+LO.CON
020110  005067  000010G                     CLR     RQB+LO.PAR
020114                                      CALL    REQNTL                  ; REQUEST NETLOADER TO LOAD PROCESS
020120  005767  000000G                     TST     RQB                     ; TEST STATUS BLOCK
020124  100004                              BPL     10$
020126                                      NTIERR  17                      ; FAILED TO LOAD A PROCESS
020136                              10$:    RETURN
```

```
                                                                              J  1
VNCP          CREATED BY  MACRO  ON 29-JUN-85 AT 02:44     PAGE 1
SYMBOL CROSS REFERENCE                                     CREF    04.00
SYMBOL  VALUE             REFERENCES

ACCTTL     001512 R      #5-171     5-171      7-473
ADDRS      000102 R      #5-131    *7-753      7-756     *7-768      7-787      7-789      7-792      7-960      7-962
                          7-965     7-1011     7-1014     7-1017     7-1072     7-1075     7-1078
ARAMSK   = ******   GX    7-785     7-957      7-1008     7-1069
AREAN      000100 R      #5-130    *7-752     *7-767      7-777     *7-786      7-793      7-949     *7-958      7-966
                          7-1000   *7-1009     7-1018     7-1061    *7-1070     7-1079
ASCII      000064 R      #5-123     8-1291     8-1300
BDGRP      002047 R      #5-178     5-178      7-396      7-584
BDHELT     002110 R      #5-179     5-179      7-574
BDLIST     002144 R      #5-180     5-180      7-655
CHKLOG     006366 R       7-450     7-638      7-1197     7-1233    #12-1428
CKEVE      000452 R      #7-290     7-356      7-373
CKEVEB     000536 RG     #7-318
CKEVEC     000566 RG     #7-333
CKEVEN     000640 RG     #7-352
CKEVER     000666 RG     #7-364
CM$SHO   = ******   GX    7-848
CNUMTL     000357 R      #5-152     5-152      7-456
CRLF       000106 R      #5-143     5-144      6-220
DTEATL     002257 R      #5-183     5-183      7-522
D$END    = 000104         7-549     7-781      7-953      7-1004     7-1065
D$LNUM     000014         7-551     7-784      7-956      7-1007     7-1068
D.NAM    = ******   GX    8-1318
D.PCB    = ******   GX    8-1317
D.UCB    = ******   GX    7-910
D.UCBL   = ******   GX    7-908
D.UNIT   = ******   GX    7-901     7-903      7-906
EBUF     = 000014 R      #5-107     5-108     *6-199     *6-243     *6-251      6-252     *7-268     *7-285     *7-396
                         *7-413    *7-425     *7-438     *7-456     *7-473     *7-4[illegible]  *7-517     *7-522     *7-543
                         *7-564    *7-574     *7-584     *7-655     *7-667     *7-6[illegible]  *7-690     *7-701     *7-712
                         *7-723    *7-734     *7-746     *7-772     *7-797     *7-[illegible]   *7-825     *7-842     *7-859
                         *7-869    *7-881     *7-913     *7-926     *7-942     *7-[illegible]9  *7-1021    *7-1083    *7-1116
                         *7-1127   *7-1148    *7-1185    *7-1208    *7-1209    *7-1221    *7-1244    *7-1245    *7-1248
                         *7-1262   *7-1280
ELEN     = 000016 R      #5-108    *6-243     *6-251     *7-268     *7-285     *7-396     *7-413     *7-425     *7-438
                         *7-456    *7-473     *7-490     *7-517     *7-522     *7-543     *7-564     *7-574     *7-584
                         *7-655    *7-667     *7-679     *7-690     *7-701     *7-712     *7-723     *7-734     *7-746
                         *7-772    *7-797     *7-809     *7-825     *7-842     *7-859     *7-869     *7-881     *7-913
                         *7-926    *7-942     *7-969     *7-1021    *7-1083    *7-1116    *7-1127    *7-1148    *7-1185
                         *7-1208   *7-1209    *7-1221    *7-1244    *7-1245    *7-1248    *7-1262    *7-1280
EMSG       000000 R      #5-106     5-107      10-1374
EOJ        000320 R       6-250    #6-254
ERRHDL     006304 R       6-243     6-251      7-268      7-285      7-396      7-413      7-425      7-438      7-456
                          7-473     7-490      7-517      7-522      7-543      7-564      7-574      7-584      7-655
                          7-667     7-679      7-690      7-701      7-712      7-723      7-734      7-746      7-772
                          7-797     7-809      7-825      7-842      7-859      7-869      7-881      7-913      7-926
                          7-942     7-969      7-1021     7-1083     7-1116     7-1127     7-1148     7-1185     7-1208
                          7-1209    7-1221     7-1244     7-1245     7-1248     7-1262     7-1280    #10-1374
ERRLID     005654 R       7-445     7-1192     7-1206     7-1228     7-1242    #7-1248     7-1274     7-1278
EVEBEG   = 000062 R      #5-118    *7-326      7-365
FL.CNT   = ******   GX    7-1267    7-1269
FNDDCB     006146 R       7-898    #8-1313
```

```
                                                                              K  1
VNCP          CREATED BY  MACRO  ON 29-JUN-85 AT 02:44     PAGE 2
```

```
        .TITLE  VNP - VIRTUAL NETWORK COMMAND PROCESSOR
        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       VIRTUAL NETWORK LOADER (VNP) MAIN CONTROL LOOP
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;


```

FUNCTION CODE DISPATCH TABLE

```
    449 001160  000000G                                      .WORD   $VALI           ; 17 - CLEAR ALIAS
    450 001162  000000G                                      .WORD   $VLOG           ; 20 - SET LOGGING
    451 001164  000000G                                      .WORD   $VLOG           ; 21 - CLEAR LOGGING
    452 001166  000000G                                      .WORD   $VOBJ           ; 22 - SET OBJECT
    453 001170  000000G                                      .WORD   $VOBJ           ; 23 - CLEAR OBJECT
    454 001172  000000G                                      .WORD   $DSYS           ; 24 - SHOW SYSTEM
    455 001174  000000G                                      .WORD   $DLIN           ; 25 - SHOW LINE
    456 001176  000000G                                      .WORD   $DPRO           ; 26 - SHOW PROCESS
    457 001200  000000G                                      .WORD   $DNOD           ; 27 - SHOW NODE
    458 001202  000000G                                      .WORD   $DALI           ; 30 - SHOW ALIAS
    459 001204  000000G                                      .WORD   $DLOG           ; 31 - SHOW LOGGING
    460 001206  000000G                                      .WORD   $DOBJ           ; 32 - SHOW OBJECT
    461 001210  002024'                                      .WORD   $SCIR           ; 33 - SET CIRCUIT
    462 001212  000000                                       .WORD   0               ; 34 - CLEAR CIRCUIT
    463 001214  000000G                                      .WORD   $DCIR           ; 35 - SHOW CIRCUIT
    464 001216  000000G                                      .WORD   $SMOD           ; 36 - SET MODULE
    465 001220  000000G                                      .WORD   $DMOD           ; 37 - SHOW MODULE
```

```
                                                                                           J  4
VNP - VIRTUAL NETWORK COMMAND P MACRO V05.03b Saturday 29-Jun-85 02:45  Page 20-2
Symbol table

$ALMHC= ****** GX       $DLOG = ****** GX       $PDVA = ****** GX       $VACT = ****** GX       $VSER = ****** GX
$ALTLC= ****** GX       $DMOD = ****** GX       $PDVTA= ****** GX       $VALI = ****** GX       $VTAD = ****** GX
$BFR  = ****** GX       $DNOD = ****** GX       $QUAL = ****** GX       $VCOST= ****** GX       $VTST = ****** GX
$BIAS = ****** GX       $DOBJ = ****** GX       $RADDR= ****** GX       $VDEA = ****** GX       $VXIT = ****** GX
$CACH = ****** GX       $DPRO = ****** GX       $ROL  = ****** GX       $VENA = ****** GX       $XFER = ****** GX
$CCEX = ****** GX       $DSYS = ****** GX       $RSNTI= ****** GX       $VFLAG= ****** GX       $XLIN = ****** GX
$CKT  = ****** GX       $EROUT= ****** GX       $SAVAL= ****** GX       $VHELP= ****** GX       $XPRO = ****** GX
$CLFLG= ****** GX       $FLAGS= ****** GX       $SCEX = ****** GX       $VHTM = ****** GX       $$    = 177777
$CLIN = ****** GX       $FUNC = ****** GX       $SCIR   002024R         $VINI = ****** GX       $$CSR = ****** GX
$CLSAV= ****** GX       $GETRV= ****** GX       $SLIN   002236R         $VLOG = ****** GX       $$DEV = ****** GX
$CNOD = ****** GX       $MISS = ****** GX       $SLTA = ****** GX       $VLTM = ****** GX       $$HTIM= ****** GX
$CPRO = ****** GX       $NAME = ****** GX       $SLTMA= ****** GX       $VNCP = ****** GX       $$PRI = ****** GX
$CSRLD= ****** GX       $NLCFG= ****** GX       $SLTNM= ****** GX       $VNP    000104RG        $$VECT= ****** GX
$DALI = ****** GX       $NLDEV= ****** GX       $SMOD = ****** GX       $VNTI = ****** GX       $$$   = 001500R
$DCIR = ****** GX       $NLRQB= ****** GX       $SNOD = ****** GX       $VOBJ = ****** GX       $$$$  = 001546R
$DDMPV= ****** GX       $OPTON= ****** GX       $SPRO   001614R         $VOWN = ****** GX       .MGMGE= ****** GX
$DLCPV= ****** GX       $PANIC  000130RG        $TMP  = ****** GX       $VRQB   000450RG        .PLGTH= ****** GX
$DLIN = ****** GX

. ABS.  000106    000   (RW,I,GBL,ABS,OVR)
        002572    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13761 Words  ( 54 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:26.51
SY:VNP.V2,[132,134]VNP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VNP
```

```
                                                                 K  4
VNP         CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 1
```

```
                                                                     J  5
VNSD - VNP SYMBOL DEFINITIONS   MACRO V05.03b Saturday 29-Jun-85 02:46  Page 5-2
Symbol table

TS$SUP= 000010        US.LAB= 000004        U.BUF   000024        U.SHST= 000076        U.VSER= 000120
TS$XHR= 000004        US.MDE= 000002        U.CBF = 000032        U.SHUN= 000074        U.2MED= 000070
T$$KMG= 000000        US.MDM= 000020        U.CLI   177772        U.SPC = 000036        U2.AT.= 000020
T$$MIN= 000000        US.MNT= 000100        U.CNT   000030        U.STS   000005        U2.CRT= 002000
T2$CLI= 000001        US.OFL= 000001        U.COTQ  000030        U.ST2   000007        U2.DH1= 100000
T2$FMP= 000002        US.PUB= 000004        U.CTCB  000026        U.SUB = 000036        U2.DJ1= 040000
UC.ALG= 000200        US.PWF= 000010        U.CTL   000004 G      U.TCHP  000042        U2.DZ1= 000100
UC.ATT= 000010        US.RED= 000002        U.CTYP  000052        U.TCVP  000043        U2.ESC= 001000
UC.KIL= 000004        US.SHR= 000001        U.CW1   000010        U.TFLK  000040        U2.HFF= 010000
UC.LGH= 000003        US.SPU= 000002        U.CW2   000012        U.TFRQ  000036        U2.HLD= 000040
UC.NPR= 000100        US.UMD= 000010        U.CW3   000014        U.TIXL  000060        U2.LOG= 000400
UC.PWF= 000020 G      US.VV = 000001        U.CW4   000016        U.TLPP  000053        U2.LWC= 000001
UC.QUE= 000040        US.WCK= 000010        U.CYL = 000104        U.TMTI  000047        U2.L3S= 000004
UD.UNS= 000000        UU.ABO= 000400        U.DCB   000000 G      U.TRCK= 000100        U2.L8S= 010000
UD.160= 000004        UU.ATN= 000100        U.FCDE= 000042        U.TSTA  000026        U2.NEC= 004000
UD.200= 000001        UU.AVN= 000004        U.FNUM= 000040        U.TST5  000054        U2.PRV= 000010
UD.556= 000002        UU.GUS= 000010        U.GRP = 000102        U.TST6  000056        U2.RMT= 020000
UD.625= 000005        UU.IOS= 002000        U.KCSR= 000032        U.TTAB  000050        U2.R04= 100000
UD.8K = 000006        UU.ONL= 000020        U.KCS6= 000034        U.TTYP  000046        U2.SCS= 000004
UD.800= 000003        UU.RCT= 000002        U.LUIC  177774        U.TUX   000024        U2.SLV= 000200
UM.CLI= 000036        UU.RDY= 000200        U.MEDI= 000040        U.UHVR= 000107        U2.VT5= 000002
UM.DSB= 000200        UU.SER= 000001        U.MLUN= 000050        U.UIC   000044        U2.7CH= 010000
UM.NBR= 000400        UU.SIO= 001000        U.MUP   177772        U.UNFL= 000052        U3.OPA= 100000
UM.OVR= 000001        UU.SPC= 000040        U.OWN   177776        U.UNIT  000006        U3.PAR= 040000
US.ABO= 000001        U.ACB   000062        U.RBNS= 000112        U.UNSZ= 000114        U3.UPC= 020000
US.BSP= 000002        U.ACP = 000032        U.RCTC= 000113        U.UNTI= 000060        U4.CR = 000100
US.BSY= 000200        U.ADMA  000066        U.RCTS= 000110        U.USVR= 000106        V$$CTR= 001000
US.FOR= 000040        U.AFLG  000064        U.RED   000002 G      U.UTIL= 000036        X$$DBT= 000000
US.FRK= 000002        U.ATT   000022        U.RED2  000034        U.VCB = 000034        $LBXL  = 000340
US.KPF= 000001        U.BPKT= 000044        U.SCB   000020 G

. ABS.  177776    000    (RW,I,GBL,ABS,OVR)
        000000    001    (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12913 Words  ( 51 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.11
SY:VNSD.V2,[132,134]VNSD/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]VNSD
```

```
                                                       K  5
VNSD       CREATED BY  MACRO  ON 29-JUN-85 AT 02:46      PAGE 1
```

```
                                          ;
000370  010167  000000G                   110$:  MOV     R1,SUBPAR              ; DO WE HAVE PCB ADDRESS ?
000374  001004                                   BNE     120$                   ; YES
000376                                           NTIERR  2                      ; UNABLE TO CREATE SUB-PARTITION
                                          ;
                                          ; FINISH PARTITION ALLOCATION
                                          ;
000406  032767  000020  000000G           120$:  BIT     #FE.PLA,$FMASK         ; PLAS SYSTEM?
000414  001425                                   BEQ     125$                   ; NO IF EQ
000416                                           GETRV   R1,.PLGTH              ; RETRIEVE SUB PCB
000434  016702  000000G                          MOV     .PPRO,R2               ; PROTECTION WORD OFFSET IN PCB
000440  012762  177774  000000G                  MOV     #^C<3>,$BFR(R2)        ; ALLOW ONLY SYSTEM READ/
000446  005722                                   TST     (R2)+                  ; POINT AT ATTACH LISTHEAD
000450  005062  000000G                          CLR     $BFR(R2)               ; CLEAR FIRST LIST POINTER
000454  010146                                   MOV     R1,-(SP)               ; ADDR OF SUB PCB
000456  060216                                   ADD     R2,(SP)                ; PLUS OFFSET TO FIRST POINTER
000460  012662  000002G                          MOV     (SP)+,$BFR+2(R2)       ; POINT LAST LISTHEAD TO FIRST
000464                                           PUTRC                          ; WRITE PCB BACK
                                          ;
                                          ; ALLOCATE UMRS IF IT'S AN 11/70.  NOTE THAT ON MULTI-PROCESSOR SYSTEMS,
                                          ; EACH CPU HAS ITS OWN POINTER TO ITS UMR BLOCK, AND THE UMR BLOCK POINTER
                                          ; IN THE NTL HOME BLOCK IS NOT USED.
                                          ;
000470                                    125$:  .IF     DF,R$$MPL

                                                 MOV     .MPRO,R2               ; MULTI PROCESSOR SYSYTEM ?
                                                 BMI     128$                   ; IF NO - BRANCH
                                          126$:  MOV     R2,R3                  ; COPY THE PROCESSOR NUMBER
                                                 DEC     R3                     ; CHANGE OFFSET TO ZERO
                                                 ASL     R3                     ; MAKE INTO A WORD INDEX
                                                 ADD     .K6TAB,R3              ; POINT INTO CPU AREA TABLE
                                                 SWTCHI  #$CPBIA                ; SET TO ''GET'' INTO CPU BIAS WORD
                                                 GETRV   R3,#2                  ; GET THE CPU AREA'S BIAS
                                                 SWTCHI                         ; RESET INPUT BUFFER
                                                 MOV     R2,-(SP)               ; SAVE THE CPU COUNT
                                                 CALL    $UMRAL                 ; ALLOCATE THE UMR BLOCK FOR THIS CPU
                                                 MOV     (SP)+,R2               ; RECOVER THE CPU COUNT
                                                 DEC     R2                     ; NEXT CPU
                                                 BGT     126$                   ; BR IF MORE TO GO
                                                 BR      130$                   ; ALL UMR BLOCKS NOW ALLOCATED
                                                 .ENDC

000470                                    128$:  CALL    $UMRAL                 ; ALLOCATE THE UMR BLOCK
                                          ;
                                          ; LOAD THE COMM EXEC
                                          ;
000474  012767  000001  000000G           130$:  MOV     #LS.CEX,RQB            ; SET FUNCTION CODE
000502  016767  000000G 000002G                  MOV     PSTART,RQB+2           ; SET NETLDR HOME BLOCK ADDRESS
000510                                           CALL    REQNTL                 ; REQUEST NETLDR
000514  005767  000000G                          TST     RQB                    ; DID COMM EXEC LOAD SUCCEED?
000520  100004                                   BPL     132$                   ; IF PL, YES
000522                                           NTIERR  4                      ; FAILED TO LOAD CEX
000532                                    132$:
000532                                           SWTCHO                         ; RESET BUFFERS
000540  016767  000000G 000000G                  MOV     SUBPAR,$NTLHB          ; COPY POOL PCB ADDRESS
000546  016767  000000G 000002G                  MOV     UMRPTR,$NTLHB+2        ; COPY UMR BLOCK ADDRESS
000554  016702  000000G                          MOV     PSTART,R2              ; GET POOL STARTING BLOCK NUMBER
```

```
                                                .SBTTL DEALLOCATE RESOURCES

                                        ;+
                                        ;       **-$DALCT       - DEALLOCTE ALLOCATED RESOURCES
                                        ;
                                        ; INPUTS:
                                        ;       $CCBLH  - CCB LISTHEAD
                                        ;       $RDBLH  - RDB LISTHEAD
                                        ;       $SDBLH  - SDB LISTHEAD
                                        ;
                                        ;
                                        ;-

003200                                  $DALCT::
003200  032767  020000  000000G                 BIT     #FE.CEX,$FMASK          ; HAS COMM EXEC BEEN LOADED?
003206  001442                                  BEQ     20$                     ; IF EQ, NO
                                        ;
                                        ; FIRST RETURN CCBS
                                        ;
003210  016701  000000G                         MOV     $CCBSZ,R1               ; ALL CCBS ARE THIS SIZE

003214  016700  000000G                 5$:     MOV     $CCBLH,R0               ; GET ADDRESS OF NEXT CCB
003220  001415                                  BEQ     10$
003222                                          GETRV   R0,#2                   ; GET IMAGE INTO BUFFER
003240  016767  000000G 000000G                 MOV     $BFR,$CCBLH             ; MAINTAIN QUEUE LISTHEAD
003246                                          CALL    $DEA16                  ; DEALLOCATE NODE
003252  000760                                  BR      5$                      ; LOOP FOR NEXT NODE
                                        ;
                                        ; RETURN RDB'S CCBS
                                        ;
003254  005067  000000G                 10$:    CLR     $RDBLH                  ; RESTORE RDB LISTHEAD
003260  005067  000002G                         CLR     $RDBLH+2                ;   TO INITIAL VALUE

003264  005067  000000G                 15$:    CLR     $SDBLH                  ; RESTORE SDB LISTHEAD
003270  005067  000002G                         CLR     $SDBLH+2                ;  TO INITIAL VALUE
                                        ;
                                        ; UNLOAD COMM EXEC
                                        ;
003274  012767  000004  000000G                 MOV     #LX.CEX,RQB             ; CLEAR CEX FUNCTION CODE
003302                                          CALL    REQNTL                  ; REQUEST NTL
003306  005767  000000G                         TST     RQB                     ; SUCCESS?
003312  100024                                  BPL     30$                     ; IF PL, YES - NOTHING ELSE TO DO
                                        ;
                                        ; UNLOAD NETWORK POOL PARTITION
                                        ;
003314  016700  000000G                 20$:    MOV     SUBPAR,R0               ; GET PCB ADDRESS
003320  001407                                  BEQ     25$                     ; WAS NOT ALLOCATED
003322                                          CALL    $XPAR                   ; DEALLOCATE IT

003326  016700  000000G                         MOV     SBPCB,R0                ; GET SERVER BASE BUFFER PCB ADDRESS
003332  001402                                  BEQ     25$                     ; WAS NOT ALLOCATED
003334                                          CALL    $XPAR                   ; DEALLOCATE IT
                                        ;
                                        ; DE-ALLOCATE UMR BLOCK
                                        ;
003340                                  25$:    .IF     DF,R$$MPL
                                                TST     .MPRO                   ; MULTIPROCESSOR SYSTEM ?
                                                BMI     28$                     ; IF YES - BRANCH
```

```
   1439 004350  060167  000000G                 ADD     R1,NTPSIZ                       ; UPDATE TOTAL PARTITION SIZE
   1440
   1441 004354  016700  000072'                 MOV     ECLNOD,R0                       ; STORE ECL NUMBER OF NODE COUNTERS
   1442 004360  012701  000050                  MOV     #E$NLEN,R1                      ; STORE NODE COUNTER SIZE
   1443 004364                                  CALL    $MUL                            ; CALCULATE TOTAL
   1444 004370  062701  000077                  ADD     #77,R1                          ; ROUND UP TO NEXT BLOCK
   1445 004374  042701  000077                  BIC     #77,R1                          ; ...
   1446
   1447                 000006                  .REPT   6
   1448                                         ASR     R1                              ; CONVERT TO BLOCKS
   1449                                         .ENDM
   1450 004414  060167  000000G                 ADD     R1,NONMAP                       ; ALLOCATE MORE NON-UNR MAPPED POOL
   1451 004420  060167  000000G                 ADD     R1,NTPSIZ                       ; UPDATE TOTAL SIZE
   1452
   1453 004424                                  RETURN
   1454
   1455 004426  005767  000000G         .NOMAP: TST     .MGEXT                          ; 11/70 TARGET SYSTEM?
   1456 004432  100003                          BPL     10$                             ; IF PL, YES
   1457 004434  012767  000002  000000G         MOV     #2,UMRFLG                       ; ELSE DISABLE UMR'S
   1458 004442  066767  000000G 000000G 10$:    ADD     .PNUMB,NONMAP                   ; AMOUNT OF NON-UMR-MAPPED POOL
   1459 004450                                  RETURN
   1460
   1461 004452  012746  000000G         .NAM1:  MOV     #NTPNAM,-(SP)
   1462 004456  000402                          BR      .NAM
   1463 004460  012746  000000G         .NAM2:  MOV     #NTPPAR,-(SP)
   1464 004464  012700  000026'         .NAM:   MOV     #NAMBUF+4,R0                    ; COPY TPARS STRING TO LOCAL BUFFER
   1465 004470  012710  020040                  MOV     #"  ,(R0)                       ; AND FILL OUT TO SIX CHARACTERS
   1466 004474  011040                          MOV     (R0),-(R0)                      ; WITH BLANKS
   1467 004476  011040                          MOV     (R0),-(R0)
   1468 004500  016701  000000G                 MOV     .PSTPT,R1
   1469 004504  016702  000000G                 MOV     .PSTCN,R2
   1470 004510  022702  000006                  CMP     #6,R2                           ; DON'T COPY MORE THAN SIX
   1471 004514  103002                          BHIS    10$
   1472 004516  012702  000006                  MOV     #6,R2
   1473 004522  112120                  10$:    MOVB    (R1)+,(R0)+
   1474 004524  005302                          DEC     R2
   1475 004526  001375                          BNE     10$
   1476 004530  012700  000022'                 MOV     #NAMBUF,R0                      ; NOW CONVERT TO RAD50
   1477 004534  010701                          MOV     PC,R1                           ; ALLOWING PERIODS
   1478 004536                                  CALL    $CAT5
   1479 004542  011602                          MOV     (SP),R2
   1480 004544  010122                          MOV     R1,(R2)+
   1481 004546  010216                          MOV     R2,(SP)
   1482 004550  010701                          MOV     PC,R1
   1483 004552                                  CALL    $CAT5
   1484 004556  010136                          MOV     R1,@(SP)+
   1485 004560                                  RETURN
   1486
   1487 004562  066767  000000G 000000G .ALLC:  ADD     .PNUMB,.ALLOC                   ; INCREASE SIZE OF SINGLE-WORD POOL
   1488 004570  066767  000000G 000000G         ADD     .PNUMB,NTPSIZ                   ; INCREASE SIZE OF POOL
   1489 004576                                  RETURN
   1490
   1911                                                  .SBTTL  BUF$DF PARSE TABLES
   1912                                          ;+
   1913                                          ;       STATE TRANSITIONS FOR THE 'BUF$DF' MACRO
   1914                                          ;-
   1915
   1916 006324                                   STATE$  BUFDF
   1917 006324                                    TRAN$   $NUMBR,BUFLP,,BUFNM
   1918 006324                                    TRAN$   $LAMDA,,,BUFSK
   1919 006324                                   STATE$  BUFLP
   1920 006324                                    TRAN$   <',>,BUFDF
   1921 006324                                    TRAN$   !END,$EXIT,,BUFCT
   1922 006324                                   STATE$
   2289                                                                    .SBTTL  SUB-EXPRESSION PARSE TABLES
   2290                                                            ;+
   2291                                                            ;       SUB-EXPRESSION FOR EXIT CONDITION
   2292                                                            ;-
   2293
   2294 006324                                                             STATE$  END
   2295 006324                                                              TRAN$  <';>,$EXIT
   2296 006324                                                              TRAN$  $EOS,$EXIT
   2297
   2298                                                            ;+
   2299                                                            ;       SUB-EXPRESSION TRANSITIONS FOR 'PROCESS NAME', ','
   2300                                                            ;-
   2301
   2302 006324                                                             STATE$  PRONAM
   2303 006324                                                              TRAN$   $RAD50
   2304 006324                                                             STATE$
   2305 006324                                                              TRAN$   <',>,$EXIT
   2306 006324                                                             STATE$
   2307
   2308                                                            ;+
   2309                                                            ;       SUB-EXPRESSION TRANSITIONS FOR ANGLE-BRACKETED STRINGS
   2310                                                            ;-
   2311
   2312 006324                                                             STATE$  STRING
   2313 006324                                                              TRAN$   RBRA,STRNG1
   2314 006324                                                              TRAN$   $ANY,STRING
   2315 006324                                                             STATE$  STRNG1
   2316 006324                                                              TRAN$   $LAMDA,$EXIT,.STRNG
   2317 006324                                                             STATE$
   2318
   2319
   2320
```

```
                                            .SBTTL  ACTION ROUTINE: 'MARKED FOR LOAD' FLAG SEEN
                                    ;+
                                    ;       **-.PROMF-'MARKED FOR LOAD' FLAG SEEN
                                    ;
                                    ;       ONE OF THE PROCESS DEFINITION MACROS HAS BEEN FOUND TO CONTAIN
                                    ;       A 'MARKED FOR LOAD' REFERENCE. SCAN THE PROCESS DESCRIPTOR
                                    ;       VECTORS FOR THE CORRECT PROCESS AND SET THE 'MARKED FOR LOAD'
                                    ;       BIT IN THE FLAG BYTE.
                                    ;-
                                    ;
006520  016700  000000G             .PROMF: MOV     $PDVTA,R0               ; GET POINTER TO PROCESS INDEX TABLE
006524  016701  000000G                     MOV     $PDVNM,R1               ; GET # OF ENTRIES TO SCAN
006530  012002                      10$:    MOV     (R0)+,R2                ; GET POINTER TO A PDV
006532  001404                              BEQ     20$                     ; IGNORE ZERO ENTRIES
006534  026762  000000G 000004              CMP     PDVNAM,Z.NAM(R2)
006542  001403                              BEQ     30$                     ; IS THIS THE CORRECT PROCESS?
006544  005301                      20$:    DEC     R1
006546  001370                              BNE     10$                     ; NO ... KEEP SCANNING
006550                                      RETURN
006552  152762  000010  000010      30$:    BISB    #ZF.MFL,Z.FLG(R2)
006560                                      RETURN
```

```
   3066 010724  156767  000000G 000045G            BISB    .PNUMB,D$DELF+$BFR ; STORE DELAY FACTOR
   3067 010732                                     PUTRC                      ; UPDATE HOME BLOCK
   3068 010736                                     RETURN
   3069
   3070 010740                             101$:   NTIERR  36                 ; ILLEGAL VALUE
   3071
   3072 010750  105767  000001G            DELW:   TSTB    .PNUMB+1           ; > 255. ?
   3073 010754  001016                             BNE     101$               ; IF YES - BRANCH
   3074 010756                                     GETRV   $DECPT,#D$END      ; GET HOME BLOCK ADDRESS
   3075 010776  156767  000000G 000046G            BISB    .PNUMB,D$DELW+$BFR ; STORE DELAY WEIGHT
   3076 011004                                     PUTRC                      ; UPDATE HOME BLOCK
   3077 011010                                     RETURN
   3078
   3079 011012                             101$:   NTIERR  37                 ; ILLEGAL VALUE
   3080
   3081 011022  105767  000001G            IPL:    TSTB    .PNUMB+1           ; > 255. ?
   3082 011026  001016                             BNE     101$               ; IF YES - BRANCH
   3083 011030                                     GETRV   $DECPT,#D$END      ; GET HOME BLOCK ADDRESS
   3084 011050  156767  000000G 000051G            BISB    .PNUMB,D$IPL+$BFR  ; STORE INPUT PACKET LIMITER
   3085 011056                                     PUTRC                      ; UPDATE HOME BLOCK
   3086 011062                                     RETURN
   3087
   3088 011064                             101$:   NTIERR  40                 ; ILLEGAL VALUE
   3089
   3090
   3091 011074  105767  000001G            LST:    TSTB    .PNUMB+1           ; >255. ?
   3092 011100  001016                             BNE     101$               ; IF YES - BRANCH
   3093 011102                                     GETRV   $DECPT,#D$END      ; GET HOME BLOCK ADDRESS
   3094 011122  156767  000000G 000047G            BISB    .PNUMB,D$LST+$BFR  ; STORE LINK SERVICE THRESHOLD
   3095 011130                                     PUTRC                      ; UPDATE HOME BLOCK
   3096 011134                                     RETURN
   3097
   3098 011136                             101$:   NTIERR  41                 ; ILLEGAL VALUE
   3099
   3100
   3101 011146  105767  000001G            RETF:   TSTB    .PNUMB+1           ; > 255. ?
   3102 011152  001016                             BNE     101$               ; IF YES - BRANCH
   3103 011154                                     GETRV   $DECPT,#D$END      ; GET HOME BLOCK
   3104 011174  156767  000000G 000050G            BISB    .PNUMB,D$RETF+$BFR ; STORE RETRANSMIT FACTOR
   3105 011202                                     PUTRC                      ; UPDATE HOME BLOCK
   3106 011206                                     RETURN
   3107
   3108 011210                             101$:   NTIERR  45                 ; ILLEGAL RETRANSMIT FACTOR
   3109
   3110 011220                             SEG:    GETRV   $DECPT,#D$END      ; GET HOME BLOCK
   3111 011240  016767  000000G 000036G            MOV     .PNUMB,D$SEG+$BFR  ; STORE ECL SEGMENT SIZE
   3112 011246                                     PUTRC                      ; UPDATE HOME BLOCK
   3113 011252                                     RETURN
   3114
   3115
```

```
                                        .SBTTL  ACTION ROUTINE: SET UP FILTER CONTROL BLOCK

                                ;+
                                ;
                                ; E.CON - SINK TYPE CONSOLE FOUND
                                ; E.FIL - SINK TYPE FILE FOUND
                                ; E.MON - SINK TYPE MONITOR FOUND
                                ;
                                ; INPUTS:
                                ;       CLASS - EVENT CLASS
                                ;       EVENT - EVENT MASKS
                                ;
                                ;-
                                        .ENABL  LSB
012676  052767  000001  000044' E.CON:  BIS     #FF.CON,FLTFLG          ; INDICATE EVENT DESTINED FOR CONSOLE
012704  000407                          BR      5$
012706  052767  000002  000044' E.FIL:  BIS     #FF.FIL,FLTFLG          ; INDICATE EVENT DESTINED FOR FILE
012714  000403                          BR      5$
012716  052767  000004  000044' E.MON:  BIS     #FF.MON,FLTFLG          ; INDICATE EVENT DESTINED FOR MONITOR

012724  012701  000016          5$:     MOV     #F.LEN,R1               ; GET LENGTH OF FILTER CONTROL BLOCK
012730                                  CALL    $ALPOL                  ; TRY TO ALLOCATE FILTER BLOCK
012734  103457                          BCS     101$                    ; BR IF COULDN'T
012736  012701  000000G                 MOV     #$BFR,R1                ; POINT AT I/O BUFFER
012742  016761  000042' 000002          MOV     CLASS,F.CLS(R1)         ; STORE EVENT CLASS
012750  052761  000016  000002          BIS     #F.LEN,F.CLS(R1)        ; SET LENGTH OF FILTER BLOCK
012756  016761  000046' 000004          MOV     EVENT+0,F.EVT+0(R1)     ; STORE EVENT TYPE MASKS
012764  016761  000050' 000006          MOV     EVENT+2,F.EVT+2(R1)     ; ...
012772  016761  000052' 000010          MOV     EVENT+4,F.EVT+4(R1)     ; ...
013000  016761  000054' 000012          MOV     EVENT+6,F.EVT+6(R1)     ; ...
013006  016761  000044' 000014          MOV     FLTFLG,F.FLG(R1)        ; STORE FLAGS
013014  010067  000004'                 MOV     R0,CURUNM               ; SAVE UNMAPPED ADDRESS
013020                                  CEACC$  R0                      ; CONVERT TO MAPPED ADDRESS
013030                          10$:    PUTAD   R0,#F.LEN               ; WRITE OUT FILTER BLOCK
                                ;
                                ; INSERT ENTRY INTO LIST
                                ;
013046  016701  000042'                 MOV     CLASS,R1                ; GET CLASS NUMBER
013052  016700  000004'                 MOV     CURUNM,R0               ; GET UNMAPPED ADDR OF BLOCK TO INSERT
013056                                  CALL    FINFIL                  ; FIND FILTER BLOCK
013062  012702  000000G                 MOV     #$FILHD,R2              ; POINT TO LISTHEAD
013066                                  CALL    QINS                    ; INSERT ENTRY INTO LIST
013072                                  RETURN
013074                          101$:   NTIERR  23                      ; FILTER BLOCK ALLOCATION FAILURE
                                        .DSABL  LSB
```

```

                                        ;+
                                        ; SER$DF ACTION ROUTINS
                                        ;
                                        ; INPUTS -
                                        ;       .PNUMB - NUMERICAL VALUE PARSED
                                        ;       .PSTPT - PARSED STRING ADDRESS
                                        ;       .PSTCN - STRING SIZE
                                        ;
                                        ; OUTPUTS -
                                        ;       PARAMETERS COPIED INTO NTLCF2
                                        ;-
014304  016767  000000G 000154' SERUNI::MOV     .PNUMB,CIRNUM   ; STORE UNIT NUMBER
014312                                  RETURN

014314  016767  000000G 000156' SERMUL::MOV     .PNUMB,CIRMUL   ; STORE TRID ADDRESS
014322                                  RETURN

014324  016767  000000G 000254' SERMUX::MOV     .PNUMB,CIRUNT   ; STORE UNIT NUMBER
014332                                  RETURN

014334  016700  000000G         NOAR::  MOV     .PNUMB,R0       ; STORE NODE AREA
014340  012701  002000                  MOV     #1024.,R1       ; POSITION AREA ADRESS
014344  070100                          MUL     R0,R1           ; ...
014346  010167  000112'                 MOV     R1,SERNOD       ; SAVE NODE AREA
014352                                  RETURN

014354  056767  000000G 000112' NOAD::  BIS     .PNUMB,SERNOD   ; SET NODE ADDRESS
014362                                  RETURN

014364  016700  000000G         HSAR::  MOV     .PNUMB,R0       ; SET HOST AREA
014370  012701  002000                  MOV     #1024.,R1       ; POSITION AREA ADDRESS
014374  070100                          MUL     R0,R1           ; ...
014376  010167  000114'                 MOV     R1,SERHST       ; SAVE HOST AREA
014402                                  RETURN

014404  056767  000000G 000114' HSAD::  BIS     .PNUMB,SERHST   ; SET HOST ADDRESS
014412                                  RETURN

014414  016700  000000G         SERDV:: MOV     .PSTPT,R0       ; GET STRING ADDRESS
014420                                  CALL    $CAT5           ; CONVERT TO RAD50
014424  010167  000120'                 MOV     R1,SERDEV       ; STORE DEVICE NAME

014430                                  RETURN

014432  016767  000000G 000132' DADDR:: MOV     .PNUMB,SERDMP   ; STORE DUMP ADDRESS
014440  016767  000000G 000134'         MOV     .PNUMH,SERDMP+2 ; ...

014446                                  RETURN

014450  016767  000000G 000136' DCT::   MOV     .PNUMB,SERDCT   ; STORE DUMP COUNT
014456                                  RETURN


```

```

                                 ;+
                                 ; SETBLK - SET LOCAL FILE NAME DESCRIPTOR BLOCK
                                 ;
                                 ; INPUTS -
                                 ;               DEVBUF - DEVICE STRING
                                 ;               UICBUF - UIC STRING
                                 ;               FILBUF - FILE NAME STRING
                                 ;
                                 ; OUTPUTS -
                                 ;               FNBUFF - SET UP IN A FILE NAME BLOCK FORMAT
                                 ;
                                 ;-

016040  010346                   SETBLK: MOV     R3,-(SP)        ; SAVE R3
016042  012700  000266'                  MOV     #FNBUFF,R0      ; GET BUFFER ADDRESS
016046  012701  000044                   MOV     #FNBSZ,R1       ; GET BYTE SIZE
016052                                   CALL    INIBUF          ; INIT LOCAL FNB BUFFER

016056  005020                           CLR     (R0)+           ; ZERO NEXT BLOCK POINTER
016060  105020                           CLRB    (R0)+           ; ZERO USAGE COUNT
016062  005001                           CLR     R1              ; INIT BYTE COUNT
016064  116701  000242'                  MOVB    DEVBUF,R1       ; GET FIELD SIZE

016070  012702  000242'                  MOV     #DEVBUF,R2      ; GET FIELD ADDRESS
016074  005003                           CLR     R3              ; INIT TOTAL STRING SIZE
016076  060103                           ADD     R1,R3           ; DETERMINE FNB SIZE
016100                                   CALL    MOVBUF          ; MOVE DEVBUF FIELD
016104  116701  000206'                  MOVB    UICBUF,R1       ; GET FIELD SIZE
016110  012702  000206'                  MOV     #UICBUF,R2      ; GET FIELD ADDRESS
016114  060103                           ADD     R1,R3           ; UPDATE FND SIZE
016116                                   CALL    MOVBUF          ; MOVE NEXT FIELD

016122  116701  000222'                  MOVB    FILBUF,R1       ; GET FIELD SIZE
016126  012702  000222'                  MOV     #FILBUF,R2      ; GET FIELD ADDRESS
016132  060103                           ADD     R1,R3           ; UPDATE FNB SIZE
016134                                   CALL    MOVBUF          ; MOVE NEXT FIELD
016140  062703  000006                   ADD     #FNBEXT,R3      ; INCLUDE FNB LINK WORD,USAGE AND LEN BYTES
016144  010367  000264'                  MOV     R3,FNBLN        ; STORE FNB LENGTH
016150  012603                           MOV     (SP)+,R3        ; RESTORE R3

016152                                   RETURN
```

```
   4867                                                    .SBTTL  REQUEST NETLOADER TO RUN
   4868                                            ;+
   4869                                            ;       **-REQNTL-REQUEST NETLOADER TO RUN
   4870                                            ;
   4871                                            ;       THE NETLOADER REQUEST BLOCK HAS BEEN CORRECTLY SET UP, SEND THE
   4872                                            ;       DATA BLOCK TO THE NETLOADER AND REQUEST IT TO RUN.
   4873                                            ;-
   4874                                            ;       OUTPUTS:
   4875                                            ;               C-BIT IF NETLDR IS NOT INSTALLED
   4876                                            ;
   4877 020140                                     REQNTL: CALL    $SAVAL                          ; SAVE ALL REGISTERS
   4878 020144  052767  000000G 000000G                    BIS     #VF.NTI,$VFLAG                  ; SET NTINIT MODE
   4879 020152  000167  000000G                            JMP     $VRQB                           ; JUMP TO ROOT TO PROCESS REQUEST
   4880 020156                                     $RSNTI::
   4881 020156  042767  000000G 000000G                    BIC     #VF.NTI,$VFLAG                  ; DON'T LEAVE BITS LYING AROUND
   4882                                                    .MCALL  VNPBPT
   4883 020164                                             VNPBPT  RNTI
   4884 020164                                             RETURN
   4885
   4886
   4887         000001                                     .END
```

```
                                                                 K  1
VNCP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 2
SYMBOL CROSS REFERENCE                                   CREF   04.00

SYMBOL  VALUE              REFERENCES

FORWR     004514 R          7-977      7-985     #7-1041
F.RSIZ  = ******   GX      *7-549     *7-781     *7-953     *7-1004    *7-1065    *8-1317
F.URBD  = ******   GX      *7-780     *7-782     *7-952     *7-954     *7-1003    *7-1005    *7-1064    *7-1066
GBUF    = 000044 R         #5-113      5-114     *6-220     *6-224     *6-230     *6-239
GETFAL    001451 R         #5-170      5-170      6-251
GETLIN    006314 R          7-1203     7-1239     7-1271    #11-1395
GLEN    = 000046 R         #5-114     *6-221     *6-225     *6-226     *6-231     *6-240
GMSG      000030 R         #5-112      5-113      6-222      6-228      6-232      6-241
IE.EOF  = ******   GX       6-249
IFALIT    002152 R          7-593     #7-627
IFLOGT    002156 R          7-590     #7-628
IFMODT    002162 R          7-610     #7-629
ILLARE    002312 R         #5-184      5-184      7-772
ILLCOS    000607 R         #5-157      5-157      7-425
ILLCSR    000455 R         #5-154      5-154      7-438
ILLINA    002514 R         #5-188      5-188      7-712
ILLINC    002403 R         #5-186      5-186      7-690
ILLMAC    001037 R         #5-162      5-162      7-667
ILLMDE    001105 R         #5-163      5-163      7-679
ILLNAD    000137 R         #5-147      5-147      7-797      7-969      7-1021     7-1083
ILLOBJ    001253 R         #5-166      5-166      7-825
ILLOUT    002561 R         #5-189      5-189      7-701
ILLPRI    000507 R         #5-155      5-155      7-881
ILLRET    002446 R         #5-187      5-187      7-723
ILLSEG    002344 R         #5-185      5-185      7-942
ILLTRI    001213 R         #5-165      5-165      7-1221
ILLUCS    001000 R         #5-161      5-161      7-926
ILLVEC    000550 R         #5-156      5-156      7-1262
IO.WLB  = ******   GX       5-106      5-112
IVDTEA    002225 R         #5-182      5-182      7-517
IVLIID    001407 R         #5-169      5-169      7-1248
I$$AS   = ******            9-1355
KEYTAB  = ******   GX       6-211
LACCTT  = 000035           #5-171      7-473
LBDGRP  = 000041           #5-178      7-396      7-584
LBDHEL  = 000034           #5-179      7-574
LBDLIS  = 000035           #5-180      7-655
LCNUMT  = 000042           #5-152      7-456
LCRLF   = 000006           #5-144      6-221
LDTEAT  = 000033           #5-183      7-522
LD.ALI  = 000030           #5-61       7-627
LD.CEX  = 000024           #5-61
LD.CIR  = 000035           #5-61
LD.LIN  = 000025           #5-61
LD.LOG  = 000031           #5-61       7-628
LD.MOD  = 000037           #5-61       7-629
LD.NOD  = 000027           #5-61
LD.OBJ  = 000032           #5-61
LD.PRO  = 000026           #5-61
LF.MTP  = ******   GX       7-1205     7-1275
LGETFA  = 000041           #5-170      6-251
LILLAR  = 000032           #5-184      7-772
```

```
                                                                 L  1
VNCP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 3
```

```
                                        .SBTTL  MACRO DEFINITIONS
                                ;****
                                ; MACRO CALLS
                                ;****

                                        .MCALL  GETRV,ERMSG$,ERRPT$,ERBLK$,VNPDF$,ASR$

                                        .IF DF  R$$MPL
                                        .MCALL  SCBDF$,KRBDF$
                                        .ENDC

                                        .ENABL  LC


000000                                  VNPDF$          ; DEFINE REQUEST BLOCK OFFSETS AND FUNCTION CODES

                                        .IF DF  R$$MPL                                                  ;[MP01]
                                        SCBDF$  <:>,,G  ; DEFINE SCB OFFSETS                            ;[MP01]
                                        KRBDF$  <:>     ; DEFINE KRB OFFSETS                            ;[MP01]
                                        .ENDC                                                           ;[MP01]


```

```
    467                                                 .SBTTL  NXTLIN - GET THE NEXT LINE IN THE SYSTEM
    468                                         ;+
    469                                         ; **-NXTLIN-GET THE NEXT LINE IN THE SYSTEM
    470                                         ;
    471                                         ; INPUTS:
    472                                         ;       R3 = NUMBER OF SLT'S LEFT TO SCAN
    473                                         ;       R5 = POINTER INTO SLT MAPPING TABLE
    474                                         ;
    475                                         ; OUTPUTS:
    476                                         ;       C-BIT = SUCCESS/FAILURE
    477                                         ;       R0 = SLT ADDRESS
    478                                         ;       R1 = ADDRESS OF DDM PDV
    479                                         ;       R5 = UPDATED TO NEXT LINE
    480                                         ;
    481                                         ;-
    482
    483 001222                                  NXTLIN:
    484 001222  000261                                  SEC                             ; ASSUME FAILURE
    485 001224  005303                                  DEC     R3                      ; ANY MORE LINES LEFT ?
    486 001226  002407                                  BLT     10$                     ; IF LT, NO
    487 001230  012500                                  MOV     (R5)+,R0                ; ELSE, GET SLT ADDRESS
    488 001232  116001  000000G                         MOVB    L.DDM(R0),R1            ; GET DDM PDV INDEX
    489 001236  066701  000000G                         ADD     $PDVTA,R1               ; POINT INTO PDV INDEX TABLE
    490 001242  011101                                  MOV     (R1),R1                 ; GET ADDRESS OF PDV
    491 001244  000241                                  CLC                             ; INDICATE SUCCESS
    492 001246                                  10$:    RETURN
```

```
                                                                        K  4
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 1
SYMBOL CROSS REFERENCE                                   CREF    04.00
SYMBOL   VALUE              REFERENCES

AIC         002470 R          17-651      19-736      19-747      #20-779
CIRTAB      002174 R          17-637     #17-670
CLEAR       001604 R          15-520      15-532     #15-551
CL.OPN  =   ******   GX       11-279
CNER1       000102 R         #6-88        12-390
ECHO        001250 R          11-290     #15-511
ECH2        001472 R          15-521     #15-542
ECH3        001546 R          15-533     #15-544
FL.DDM  =   ******   GX       20-781      20-782
FL.DLC  =   ******   GX       20-781      20-782
FL.LLC  =   ******   GX       16-579
FL.PX1  =   ******   GX       16-575      19-743
FL.PX2  =   ******   GX       17-658      19-754
FTAB        001124 R          11-293     #13-435
F.RSIZ  =   ******   GX      *20-790
I$$AS   =   ******            15-526
LDLIN       002320 R          18-696     #19-733
LD.ALI  =   000030           #5-69
LD.CEX  =   000024           #5-69
LD.CIR  =   000035           #5-69
LD.LIN  =   000025           #5-69
LD.LOG  =   000031           #5-69
LD.MOD  =   000037           #5-69
LD.NOD  =   000027           #5-69
LD.OBJ  =   000032           #5-69
LD.PRO  =   000026           #5-69
LF.RDY  =   ******   GX       17-664
LINTAB      002312 R          18-698     #18-712
LNER1       000024 R         #6-84        10-261
LNER2       000052 R         #6-86        10-262
LO.ADD      000022           #5-69
LO.ADR      000052           #5-69
LO.CLS      000002           #5-69
LO.CON      000006           #5-69
LO.COP      000010           #5-69
LO.COS      000020           #5-69
LO.CSR      000010           #5-69        5-69
LO.CTL      000032           #5-69
LO.DES      000002           #5-69        5-69
LO.DTE      000047           #5-69
LO.EVT      000004           #5-69
LO.GRO      000024           #5-69
LO.HOS      000040           #5-69
LO.HTM      000032           #5-69
LO.INA      000062           #5-69
LO.INC      000060           #5-69
LO.LDT      000046           #5-69
LO.LID      000030           #5-69
LO.LIN      000004           #5-69
LO.LTM      000034           #5-69
LO.MAC      000022           #5-69
LO.MDE      000024           #5-69
```

```
                                                                        L  4
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 2
```

```
                                                                 K  5
VNSD        CREATED BY  MACRO  ON 29-JUN-85 AT 02:46       PAGE 1
SYMBOL CROSS REFERENCE                                     CREF    04.00
SYMBOL   VALUE              REFERENCES

D.DSP      000012   G         5-61
D.NAM      000004   G         5-61
D.PCB      000034   G         5-61
D.UCB      000002   G         5-61
D.UCBL     000010   G         5-61
D.UNIT     000006   G         5-61
D.VPWF   = 000006   G         5-61
L$BBLK     000360   G         5-68
L$BEXT     000352   G         5-67
L$BFLG     000030   G         5-67
L$BHRB     000356   G         5-67
L$BLDZ     000016   G         5-67
L$BMXZ     000020   G         5-67
L$BPAR     000004   G         5-68
L$BSA      000010   G         5-67
L$BXFR     000350   G         5-68
P.BLKS     000016   G         5-91
P.BUSY     000024   G         5-88
P.IOC      000003   G         5-92
P.LNK      000000   G         5-92
P.MAIN     000012   G         5-91
P.NAM      000004   G         5-92
P.OWN      000026   G         5-91
P.PRI      000002   G         5-92
P.REL      000014   G         5-91
P.SIZE     000016   G         5-92
P.STAT     000030   G         5-91
P.SUB      000010   G         5-92
P.TCB      000026   G         5-92
R$$MPL   = ******             5-87
S.CON      000010   G         5-74
S.LHD      000000   G         5-74
TS$NHD   = 040000   G         5-67
UC.PWF   = 000020   G         5-80
U.CTL      000004   G         5-80
U.DCB      000000   G         5-80
U.RED      000002   G         5-80
U.SCB      000020   G         5-80
```

```
                                                                 L  5
VNSD        CREATED BY  MACRO  ON 29-JUN-85 AT 02:46       PAGE 2
```

```
    377 000560  016703  000000G                 MOV     PBLOX,R3                ; AND SIZE
    378 000564  005067  000000G                 CLR     $PAVL                   ; ASSUME NO FREE SPACE IN NTPOOL
    379 000570  005202                          INC     R2                      ; ADJUST FOR NETLDR DATA BLOCK
    380 000572  005303                          DEC     R3                      ; ...
    381 000574  005067  000000G                 CLR     $XBIAS                  ; ASSUME NO EXTENDED POOL
    382 000600  005067  000000G                 CLR     $XAVL                   ; ...
    383 000604  016704  000000G                 MOV     .ALLOC,R4               ; GET LENGTH OF EXTENDED POOL
    384 000610  001436                          BEQ     137$                    ; BR IF NONE
    385 000612  010246                          MOV     R2,-(SP)                ; SAVE ADDRESS OF BEGINNING OF POOL
    386 000614  060302                          ADD     R3,R2                   ; POINT TO END OF AVAILABLE POOL
    387 000616  160402                          SUB     R4,R2                   ; POINT TO BEGINNING OF EXTENDED POOL
    388 000620  010267  000000G                 MOV     R2,$XBIAS               ; STORE EXTENDED POOL BIAS
    389 000624  005267  000000G                 INC     $XAVL                   ; INITIALIZE FIRST AVAILABLE BLOCK
    390 000630  010267  000000G                 MOV     R2,$RBLCK               ; GET START OF BYTE AREA POOL FOR I/O
    391 000634  160403                          SUB     R4,R3                   ; CALCULATE LENGTH OF REMAINING POOL
    392 000636  012602                          MOV     (SP)+,R2                ; RESTORE BEGINNING OF POOL ADDRESS
    393 000640  006304                          ASL     R4                      ; CONVERT BLOCKS TO BYTES
    394 000642  006304                          ASL     R4                      ; ...
    395 000644  006304                          ASL     R4                      ; ...
    396 000646  006304                          ASL     R4                      ; ...
    397 000650  006304                          ASL     R4                      ; ...
    398 000652  006304                          ASL     R4                      ; ...
    399 000654  000241                          CLC                             ; ...
    400 000656  005067  000000G                 CLR     $BFR                    ; NO NEXT BLOCK
    401 000662  010467  000002G                 MOV     R4,$BFR+2               ; SIZE OF THIS BLOCK
    402 000666                                  PUTAD   #0,#4                   ; INITIALIZE POOL
    403 000706  005767  000000G         137$:   TST     .MGEXT                  ; 11/70 TARGET SYSTEM?
    404 000712  100440                          BMI     150$                    ; IF MI, NO
    405                                 ;
    406                                 ; 11/70 MAPPED SYSTEM
    407                                 ;
    408 000714  005067  000000G                 CLR     $NBIAS                  ; ZERO NON-UMR MAPPED PTR
    409 000720  005767  000000G                 TST     UMRFLG                  ; HAVE UMR'S BEEN DISABLED?
    410 000724  001033                          BNE     150$                    ; IF NE, YES
    411 000726  016704  000000G                 MOV     NONMAP,R4               ; GET SIZE OF NON-UMR MAPPED AREA
    412 000732  005304                          DEC     R4                      ; LESS 1 FOR NETLDR DATA BLOCK
    413 000734  003422                          BLE     140$                    ; IF NONE LEFT, SKIP IT
    414 000736  010267  000000G                 MOV     R2,$NBIAS               ; SETUP NON MAPPED PTR VALUE
    415 000742  010267  000000G                 MOV     R2,$RBLCK               ; MAP TO FIRST FREE SEGMENT
    416 000746  005067  000000G                 CLR     $BFR                    ; NO NEXT SEGMENT
    417 000752  010467  000002G                 MOV     R4,$BFR+2               ; LENGTH OF THIS SEGMENT
    418 000756                                  PUTAD   #0,#4                   ; INITIALIZE NON-MAPPED POOL
    419 000776  060402                          ADD     R4,R2                   ; ADJUST STARTING BIAS
    420 001000  160403                          SUB     R4,R3                   ; AND SIZE
    421
    422 001002                          140$:
    423
    424 001002  160267  000000G                 SUB     R2,PUMR                 ; CALCULATE PUMR VALUE          ;[MP01]
    425
    426                                         .IF NDF R$$MPL                                                  ;[MP01]
    427 001006  016767  000000G 000000G         MOV     PUMR,$PUMR              ; CALCULATE UMR ADDRESS CONSTANT
    428
    429                                         .IFF                                                            ;[MP01]
    430                                         TST     .MPRO                   ; MULTI-PROCESSOR ?             ;[MP01]
    431                                         BPL     145$                    ; IF YES - BRANCH               ;[MP01]
    432                                         SWTCHO  #PUMR                   ; SET OUTPUT BUFFER             ;[MP01]
    433                                         PUTRC   .PUMRP,#2               ; UPDATE $PUMR CELL IN LOWCR    ;[MP01]
```

```
                                        SWTCHI                          ; USE $BFR BUFFER
                                        MOV     .MPRO,R0                ; GET ADDRESS
                                        BLE     28$                     ; BR IF SINGLE PROCESSOR SYSTEM
                                26$:    MOV     R0,R2                   ; COPY PROCESSOR NUMBER
                                        MOV     R0,-(SP)                ; SAVE THE PROCESSOR NUMBER
                                        DEC     R2                      ; CHANGE NUMBER OFFSET TO ZERO
                                        ASL     R2                      ; MAKE IT INTO A WORD OFFSET
                                        ADD     .K6TAB,R2               ; POINT INTO THE CPU AREA TABLE
                                        SWTCHI  #$CPBIA                 ; SET TO READ INTO THE CPU AREA POINTER
                                        GETRV   R2,#2                   ; GET BIAS OF CPU'S OWN AREA
                                        SWTCHI  #UMRPTR                 ; SET TO READ INTO UMR BLOCK POINTER
                                        MOV     $CPBIA,$RBLCK           ; SET THE APR5 BIAS
                                        MOV     .UMRPT,R2               ; GET THE UMR BLOCK POINTER OFFSET
                                        BIS     #120000,R2              ; MAKE SURE ITS PAST APR5
                                        GETAD   R2,#2                   ; READ THE UMR BLOCK POINTER
                                        SWTCHI                          ; RESET TO DEFAULT INPUT BUFFER
                                        MOV     UMRPTR,R0               ; GET THE ADDRESS
                                        BEQ     27$                     ; BR IF NO UMR BLOCK FOR THIS CPU
                                        CLR     UMRPTR                  ; SET A LOCATION TO ZERO
                                        SWTCHO  #UMRPTR                 ; SET TO WRITE FROM IT
                                        PUTAD   R2,#2                   ; ZERO OUT THE UMR BLOCK POINTER
                                        SWTCHO                          ; RESET TO DEFAULT OUTPUT BUFFER
                                        MOV     R0,R2                   ; COPY UMR BLOCK ADDR FOR DEALLOCATION
                                        CALL    $DEUMR                  ; UNLINK THE UMR BLOCK
                                        MOV     #M.LGTH,R1              ; GET LENGTH OF UMR BLOCK
                                        CALL    $DEA16                  ; DEALLOCATE THE UMR BLOCK
                                27$:    MOV     (SP)+,R0                ; RECOVER THE CPU NUMBER
                                        DEC     R0                      ; NEXT CPU
                                        BGT     26$                     ; BR IF MORE
                                        BR      30$                     ; FINISHED
                                        .ENDC

003340  016700  000000G         28$:    MOV     UMRPTR,R0               ; GET UMR BLOCK ADDRESS
003344  001407                          BEQ     30$                     ; IF ZERO, DO NOTHING
003346  010002                          MOV     R0,R2                   ; COPY BLOCK ADDRESS
003350                                  CALL    $DEUMR                  ; DE-ASSIGN UMR'S
003354  012701  000014                  MOV     #M.LGTH,R1              ; GET BLOCK LENGTH
003360                                  CALL    $DEA16                  ; DE-ALLOCATE UMR BLOCK

003364                          30$:    RETURN
```

```

                                        ;+
                                        ;
                                        ; LLC$DF ACTION ROUTINES FOR POOL.. SIZE CALCULATION
                                        ;
                                        ;-

004600  016767  000000G 000066' XPCIR:  MOV     .PNUMB,XPTLLN           ; STORE XPT CIRCIUT COUNT
004606                                  RETURN

004610  016767  000000G 000070' ECLLK:  MOV     .PNUMB,ECLLLN           ; STORE MAX LOGICAL LINKS
004616                                  RETURN

004620  016767  000000G 000072' NODC:   MOV     .PNUMB,ECLNOD           ; STORE NUMBER OF NODE COUNTERS
004626                                  RETURN

004630  066767  000000G 000104' BRADJT: ADD     .PNUMB,RAD              ; ADJUST TOTAL ROUTER ADJACENCIES
004636                                  RETURN

```

```
                                                .SBTTL  LLC$DF,DLC$DF,DDM$DF PARSE TABLES
                                        ;+
                                        ;       STATE TRANSITIONS FOR THE 'LLC$DF', 'DLC$DF' AND 'DDM$DF' MACROS
                                        ;-

        006324                                  STATE$  PRODF
        006324                                   TRAN$   $RAD50,,.PRONM
        006324                                  STATE$
        006324                                   TRAN$   <',>
        006324                                  STATE$  PROFLG
        006324                                   TRAN$   '!,PROFLG
        006324                                   TRAN$   <',>,PRODF1
        006324                                   TRAN$   !END,$EXIT
        006324                                   TRAN$   %ZF.MFL%,PROFLG,.PROMF
        006324                                   TRAN$   $RAD50,PROFLG
        006324                                  STATE$  PRODF1
        006324                                   TRAN$   $NUMBR
        006324                                  STATE$
        006324                                   TRAN$   !END,$EXIT
        006324                                   TRAN$   <',>
        006324                                  STATE$
        006324                                   TRAN$   $NUMBR
        006324                                  STATE$
        006324                                   TRAN$   !END,$EXIT
        006324                                   TRAN$   <',>,$EXIT
        006324                                  STATE$
```

```



                                        ;+
                                        ; FIL$DF STATE TABLE
                                        ;-

 006324                                         ISTAT$  CFGST,CFGKW


 006324                                         STATE$  FILDF
 006324                                          TRAN$  $EOS,$EXIT
 006324                                          TRAN$  <',>,TER
 006324                                          TRAN$  !FILSPC,,.FLBK0

 006324                                         STATE$
 006324                                          TRAN$  $EOS,$EXIT
 006324                                          TRAN$  <',>

 006324                                         STATE$ TER
 006324                                          TRAN$ <',>,LOA
 006324                                          TRAN$  !FILSPC,,.FLBK1

 006324                                         STATE$
 006324                                          TRAN$  $EOS,$EXIT
 006324                                          TRAN$  <',>

 006324                                         STATE$  LOA
 006324                                          TRAN$  <',>,DIA
 006324                                          TRAN$  !FILSPC,,.FLBK2

 006324                                         STATE$
 006324                                          TRAN$  $EOS,$EXIT
 006324                                          TRAN$  <',>

 006324                                         STATE$  DIA
 006324                                          TRAN$  <',>,DUM
 006324                                          TRAN$  !FILSPC,,.FLBK3

 006324                                         STATE$
 006324                                          TRAN$  $EOS,$EXIT
 006324                                          TRAN$  <',>

 006324                                         STATE$ DUM
 006324                                          TRAN$  !FILSPC,,.FLBK4

 006324                                         STATE$
 006324                                          TRAN$  $EOS,$EXIT

```

```
                                            .SBTTL  ACTION ROUTINE: SLT$DF MACRO SEEN
                                    ;+
                                    ;       **-.SLTDF-SLT$DF MACRO SEEN
                                    ;
                                    ;       A DEFINTION OF ANOTHER SYSTEM LINE TABLE ENTRY HAS BEEN SEEN.
                                    ;       SINCE THE SYSTEM LINES ARE ORDERED WE CAN JUST INCREMENT THE
                                    ;       COUNT OF SYSTEM LINES SEEN SO FAR.
                                    ;-

 006562  005267  000000G            .SLTDF: INC     SLTNUM          ; MOVE ON SYSTEM LINE #
 006566  012767  177777  000036'            MOV     #-1,STAN        ; INITIALIZE STATION NUMBER
 006574                                     RETURN
```

```
                                                .SBTTL  ACTION ROUTINE: REMOTE NODE NAME SEEN
                                        ;+
                                        ; *** - .RNAME - REMOTE NODE NAME SEEN
                                        ;
                                        ; INPUTS:
                                        ;       .PSTPT = ADDRESS OF REMOTE NODE NAME
                                        ;       .PSTCN = LENGTH OF REMOTE NODE NAME
                                        ;
                                        ; OUTPUTS:
                                        ;       NAME = THE REMOTE NODE NAME
                                        ;
                                        ;-

011254  012700  000030'                 .RNAME: MOV     #NAME,R0                ; PLACE TO STORE NODE NAME
011260  012710  020040                          MOV     #"  ,(R0)               ; PAD NODE NAME WITH BLANKS
011264  012010                                  MOV     (R0)+,(R0)              ;       ...
011266  012010                                  MOV     (R0)+,(R0)              ;       ...
011270  022767  000006  000000G                 CMP     #6,.PSTCN               ; IS THE NODE NAME THE RIGHT LENGTH ?
011276  103411                                  BLO     101$                    ; IF LO, NO
011300  012700  000030'                         MOV     #NAME,R0                ; STORE THE NODE NAME
011304  016701  000000G                         MOV     .PSTPT,R1               ;       ...
011310  112120                          10$:    MOVB    (R1)+,(R0)+             ;       ...
011312  005367  000000G                         DEC     .PSTCN                  ; LOOP <.PSTCN> TIMES
011316  003374                                  BGT     10$                     ;       ...
011320                                          RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
011322  062716  000002                  101$:   ADD     #2,(SP)                 ; REJECT TRANSITION
011326                                          RETURN
```

```
                                                   .SBTTL  FINFIL - FIND PLACE TO INSERT FILTER BLOCK

                                          ;+
                                          ;
                                          ; FINFIL - FIND PLACE TO INSERT FILTER BLOCK
                                          ;
                                          ; INPUTS:
                                          ;        R1 - EVENT CLASS NUMBER
                                          ;
                                          ; OUTPUTS:
                                          ;        R0 - PRESERVED
                                          ;        R1 - UNMAPPED ADDRESS OF PREVIOUS BLOCK
                                          ;        R2 - DESTROYED
                                          ;
                                          ;-
 013104  012702  000000G                  FINFIL: MOV     #$FILHD,R2              ; POINT TO START OF LIST
 013110  011267  000000G                          MOV     (R2),$BFR               ; GET FIRST OBJECT IN LIST
 013114  010146                                   MOV     R1,-(SP)                ; SAVE CLASS NUMBER
 013116  010267  000004'                          MOV     R2,CURUNM               ; INIT UNMAPPED ADDR OF CURRENT BLOCK
 013122  016767  000004' 000002'          10$:    MOV     CURUNM,PRVUNM           ; SAVE UNMAPPED ADDRESS OF PREVIOUS
 013130  016767  000000G 000004'                  MOV     $BFR,CURUNM             ; GET NEXT BLOCK IN LIST
 013136  001417                                   BEQ     30$                     ; BR IF END OF LIST
 013140                                           CEACC$  CURUNM,R2               ; CONVERT TO MAPPED ADDRESS
 013152                                   20$:    GETAD   R2,#F.LEN               ; READ FILTER BLOCK INTO INTERNAL BUFFER
 013170  021667  000002G                          CMP     (SP),$BFR+F.CLS         ; DOES FILTER GO HERE?
 013174  101352                                   BHI     10$                     ; BR IF NO
 013176  016701  000002'                  30$:    MOV     PRVUNM,R1               ; RETRIEVE UNMAPPED ADDRESS OF PREVIOUS
 013202  005726                                   TST     (SP)+                   ; CLEAN UP STACK
 013204                                           RETURN
```

```
   3922
   3923                                          ;+
   3924                                          ; SERSLN - DETERMINE SLN/STA VALUE
   3925                                          ;
   3926                                          ; INPUTS -
   3927                                          ;        CIRNAM - DEVICE NAME (ASCII STRING)
   3928                                          ;        CIRNUM - DEVICE UNIT/CONTROLLER NUMBER
   3929                                          ;        CIRMUL - DEVICE TRIB NUMBER
   3930                                          ;
   3931                                          ; OUTPUTS -
   3932                                          ;        SERSLT - CONTAINS SLN/STA VALUE
   3933                                          ;-
   3934
   3935 014460  010346                           SERSLN::MOV     R3,-(SP)        ; SAVE REG
   3936 014462  012700  000150'                          MOV     #CIRNAM,R0      ; GET DEVICE STRING
   3937 014466                                           CALL    $CAT5           ; CONVERT TO RAD50
   3938 014472                                           CALL    FPDV            ; FIND DEVICE DDM PDV INDEX
   3939
   3940 014476  010100                                   MOV     R1,R0           ; GET DDM PDV INDEX
   3941 014500  005001                                   CLR     R1              ; GET CONTROLLER NUMBER
   3942
   3943 014502  116701  000154'                          MOVB    CIRNUM,R1       ; ...
   3944 014506  005002                                   CLR     R2              ; GET UNIT NUMBER
   3945 014510  116702  000254'                          MOVB    CIRUNT,R2       ; ...
   3946 014514                                           CALL    FSLT            ; GET SLN
   3947
   3948 014520  116767  000156' 000263'                  MOVB    CIRMUL,SERSLT+1 ; STORE STATION NUMBER
   3949 014526  012603                                   MOV     (SP)+,R3        ; RESTORE REG
   3950
   3951 014530                                           RETURN
   3952
   3953
   3954
   3955
   3956                                          ;+
   3957                                          ; SERINI - INIT SERVICE DATA BASE BUFFERS
   3958                                          ;
   3959                                          ; INPUTS -
   3960                                          ;        NONE
   3961                                          ;
   3962                                          ; OUTPUTS -
   3963                                          ;        SER$DF BUFFERS INITIALIZED
   3964                                          ;-
   3965
   3966 014532  012700  000110'                  SERINI::MOV     #SERBEG,R0      ; GET START OF BUFFERS
   3967 014536  005020                           5$:     CLR     (R0)+           ; CLEAR ONE WORD
   3968 014540  022700  000266'                          CMP     #SERFIN,R0      ; CHECK FOR END
   3969 014544  001374                                   BNE     5$              ; IF MORE - BRANCH
   3970
   3971 014546                                           RETURN
   3972
   3973
   3974
```

```
                                         .SBTTL  CRFNBK - CREATE FILENAME DESCRIPTOR BLOCK

                                 ;+
                                 ; CRFNBK - CREATE FILE NAME BLOCK
                                 ;
                                 ; INPUTS -
                                 ;               FNBUFF - LOCAL COPY OF FNB DATA
                                 ;               FNBLN  - SIZE OF THIS FNB
                                 ;
                                 ; OUTPUTS -
                                 ;               FILNAME DESCRIPTOR BLOCK CREATED
                                 ;               RO  - UNMAPPED ADDRESS OF FILENAME BLOCK
                                 ;
                                 ;-

016154                           CRFNBK: SAVRG   <R3>            ; SAVE REG
016156  016701  000264'                  MOV     FNBLN,R1        ; GET FNB SIZE
016162                                   CALL    $ALPOL          ; ALLOCATE FNB
016166  103003                           BCC     10$             ; IF SUCCESS - BRANCH

016170  000261                           SEC                     ; IF NOT - SET C-BIT
016172                                   RESRG   <R3>            ; RESTORE REG

016174                                   RETURN

016176  010003                   10$:    MOV     R0,R3           ; SAVE UNMAPPED ADDRESS
016200  012700  000000G                  MOV     #$BFR,R0
016204  016701  000264'                  MOV     FNBLN,R1        ; GET FNB LENGTH
016210  012702  000266'                  MOV     #FNBUFF,R2      ; GET LOCAL COPY OF NEW FNB DATA
016214  112220                   15$:    MOVB    (R2)+,(R0)+     ; TRANSFER DATA FROM LOCAL FNB BUFFER
016216  005301                           DEC     R1              ; COUNT ONE LESS BYTE
016220  001375                           BNE     15$             ; IF MORE - BRANCH

016222  010300                           MOV     R3,R0           ; RESTORE BLOCK ADDESS
016224                                   CEACC$  R0              ; MAP TO BLOCK
016234                                   PUTAD   R0,FNBLN        ; WRITE BLOCK TO IMAGE

016252                                   SWTCHI  #$HMBUF         ; SWITCH INPUT BUFFER
016260                                   GETRV   $DECPT,#D$END   ; READ HOME BLOCK
016300                                   SWTCHI

016306  012702  000034G                  MOV     #$HMBUF+D$FNB,R2 ; GET LISTHEAD
016312  010300                           MOV     R3,R0           ; COPY UNMAPPED BLOCK ADDRESS
016314  005712                           TST     (R2)            ; IS LIST EMPTY ?
016316  001002                           BNE     20$             ; IF NO - BRANCH
016320  010201                           MOV     R2,R1           ; COPY LISTHEAD
016322  000401                           BR      25$             ; CONTINUE
016324  011201                   20$:    MOV     (R2),R1         ; SET FIRST ENTRY AS PREVIOUS ENTRY
016326                           25$:    CALL    QINS            ; INSERT BLOCK INTO LIST

016332                                   SWTCHO  #$HMBUF         ; SWITCH OUTPUT BUFFER
016340                                   PUTRC   $DECPT,#D$END   ; WRITE HOME BLOCK TO IMAGE
016360                                   SWTCHO                  ; SET DEFAULT OUTPUT BUFFER
016366                                   RESRG   <R3>            ; RESTORE REG

016370                                   RETURN

```

```
                                                                  K 16
VNTI - VNP NETWORK INITIALIZATI MACRO V05.03b Saturday 29-Jun-85 02:46  Page 110-1
Symbol table

A        002214R    003 CE.ILN= 100350          CM.HRD= 000002          C.FLG1   000022          D$MAXV   000070
ADDDF    001262R    003 CE.LTO= 100356          CM.LIN= 000000          C.FLG2   000032          D$MLL    000040
AS.DEL=  000010         CE.MOP= 100372          CM.LOO= 000001          C.FNC    000010          D$MNOD   000041
AS.EXT=  000004         CE.NTE= 100361          CM.XLO= 000004          C.LIN    000006          D$NA     000062
AS.RED=  000001         CE.RTE= 100376          CP.DCF= 000040          C.LNK    000000          D$NBEA   000056
AS.WRT=  000002         CE.SRC= 100364          CP.HDL= 000007          C.MOD    000011          D$NBRA   000054
AT$ACL=  000100         CE.STP= 100352          CP.PS = 177400          C.NSP    000004          D$NEND=  000054
AT$ADN=  000040         CE.TME= 100354          CP.PSI= 000200          C.PRO    000042          D$NLN    000030
AT$AUP=  000020         CE.TMO= 100374          CP.XCF= 000100          C.RSV    000002          D$NN     000060
AT$CYC=  000004         CE.UNS= 100344          CP.2FR= 000030          C.STA    000007          D$OUTT   000043
AT$LV1=  000002         CFGKW   000116RG    004 CRFNBK  016154R         C.STS    000012          D$RETF   000050
AT$LV2=  000001         CFGST   001754RG    003 CRSUB   004640R         C.URM    177776          D$RNN    000002
AT$NEX=  000200         CFSFLG= ****** GX       CS.ABO= 000100          C.XACP   000004          D$RTMR   000076
AT$UP =  000010         CF.CHN= 000001          CS.BRO= 000002          C.XID    000035          D$SEG    000036
A$CIR    000003         CF.DDM= 000002          CS.BUF= 000200          C.XLEN   000044          D$SER    000032
A$LEN    000004         CF.DYN= 000004          CS.CES= 000002          C.XPLI   000040          D$SQRL   000052
A$NID    000000         CF.EIS= 000010          CS.CHN= 000010          C.XPT    000034          D$$BUG=  177514
A$TM     000003         CF.EOM= 000004          CS.CMP= 000200          C.XSVC   000042          D$$ISK=  000000
A$TMI    000002         CF.FRK= 100000          CS.DCR= 000400          C.XTC    000037          D$$L11=  000001
A$TSZ    000000         CF.HDR= 000020          CS.DEF= 000004          C.X25    000036          D$$YNC=  000000
A$TYP    000002         CF.LB = 100000          CS.DEV= 000002          D        002272R    003  D$$YNM=  000000
A$$CHK=  000000         CF.LIN= 000002          CS.DIS= 000040          DADDR    014432RG        E        002362R      003
A$$CPS=  000000         CF.LOG= 000020          CS.ENA= 000001          DCT      014450RG        ECLLK    004610R
A$$PRI=  000000         CF.MDM= 000001          CS.ENB= 000020          DECDF    001420R    003  ECLLLN   000070R      002
A$$TRP=  000000         CF.SOM= 000010          CS.ERR= 100000          DELF     010676R         ECLLNK   010340R
A.IOC    000003         CF.SYN= 000040          CS.FTL= 001000          DELW     010750R         ECLNOD   000072R      002
A.LGTH=  000014         CF.TIM= 000400          CS.HCR= 000001          DEV      017010RG        END      001720R      003
A.MPCT   000011         CF.TRN= 000100          CS.HFE= 002000          DEVBUF   000242RG   002  ENDHEX   000204RG     002
A.PCB    000012         CIRMUL  000156RG    002 CS.LST= 040000          DEVIC    002336R    003  ENDPAS   000132RG     002
A.PCBL   000000         CIRNAM  000150RG    002 CS.MTL= 004000          DEVNM    000160RG   002  ERRTAB   002036R
A.PRI    000002         CIRNUM  000154RG    002 CS.RNG= 000010          DFCNT  = ****** GX       ERR1     000336R      002
A.STAT   000010         CIRUNT  000254RG    002 CS.ROV= 000004          DFPTR  = ****** GX       ERR10    000742R      002
A.TCB    000004         CLASS   000042R     002 CS.RSN= 010000          DFSIZ  = 000010          ERR11    001002R      002
A.TCBL   000006         CL$MFL= 000010          CS.SHU= 000001          DFTBL    006404R         ERR12    001050R      002
B        002234R    003 CL$SFL= 000004          CS.SID= 000002          DIA      002040R    003  ERR13    001116R      002
BRADF    000324R    003 CL$TYP= 000001          CS.STR= 000004          DUM      002060R    003  ERR14    001154R      002
BRADJ    001154R    003 CL.AST= 004000          CS.SUC= 000001          DVSZ   = 000010 G        ERR15    001216R      002
BRADJJ   000450R    003 CL.ATT= 000200          CS.TMO= 020000          D$AMXC   000072          ERR16    001252R      002
BRADJT   004630R        CL.CAS= 020000          CS.XUR= 000004          D$AMXH   000074          ERR17    001306R      002
BRADJ1   001166R    003 CL.CMT= 000001          CTL     000060R     002 D$ANN    000000          ERR2     000370R      002
BRADJ2   000460R    003 CL.EIF= 000040          CURUNM  000004R     002 D$BRPR   000102          ERR20    001336R      002
BUFDF    000664R    003 CL.EXN= 040000          CV$MSK= 000003          D$BRTM   000100          ERR21    001372R      002
BUFLP    000676R    003 CL.EXT= 000010          CV$31 = 000001          D$DELF   000045          ERR22    001450R      002
BUFUMP=  172354         CL.IND= 000002          CV$32 = 000000          D$DELW   000046          ERR23    001510R      002
C        002252R    003 CL.LCC= 000020          CV$40 = 000002          D$END =  000104          ERR24    001566R      002
CB.CCB=  000002         CL.MU1= 000001          C$$CKP= 000000          D$FNB    000034          ERR25    001622R      002
CB.DDM=  000040         CL.MU2= 000002          C$$ORE= 000400          D$HIOR   000024          ERR26    001654R      002
CB.DLC=  000020         CL.NPR= 000400          C$$RSH= 177564          D$HOST   000022          ERR27    001676R      002
CB.RDB=  000004         CL.OPN= 000100          C.ADD   000034          D$INAC   000044          ERR3     000452R      002
CB.SDB=  000010         CL.RES= 177774          C.BID   000003          D$INCT   000042          ERR30    001732R      002
CB.SLI=  000100         CL.RNE= 000004          C.BUF   000014          D$IPL    000051          ERR31    001744R      002
CB.XLB=  000001         CL.XIT= 100000          C.BUF1  000014          D$LID    000020          ERR32    002000R      002
CC.LLC=  000200         CMODE = 140000          C.BUF2  000024          D$LNAM   000006          ERR33    002042R      002
CE.ABO=  100362         CMPSTR  015740R         C.CNT   000020          D$LNUM   000014          ERR34    002102R      002
CE.DAO=  100346         CM$SET= ****** GX       C.CNT1  000020          D$LST    000047          ERR35    002142R      002
CE.DIS=  100366         CM.CIR= 000002          C.CNT2  000030          D$MAXC   000064          ERR36    002200R      002
CE.ERR=  100370         CM.FMT= 100000          C.FLG   000022          D$MAXH   000066          ERR37    002232R      002
```

```
                                                                  L 16
VNTI - VNP NETWORK INITIALIZATI MACRO V05.03b Saturday 29-Jun-85 02:46  Page 110-2
Symbol table
```

```
                                                                 L  1
VNCP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:44      PAGE 3
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL  VALUE               REFERENCES

LILLCO  = 000030            #5-157       7-425
LILLCS  = 000032            #5-154       7-438
LILLIN  = 000045            #5-186      #5-188      7-690       7-712
LILLMA  = 000046            #5-162       7-667
LILLMD  = 000044            #5-163       7-679
LILLNA  = 000033            #5-147       7-797      7-969       7-1021      7-1083
LILLOB  = 000032            #5-166       7-825
LILLOU  = 000043            #5-189       7-701
LILLPR  = 000041            #5-155       7-881
LILLRE  = 000046            #5-187       7-723
LILLSE  = 000037            #5-185       7-942
LILLTR  = 000040            #5-165       7-1221
LILLUC  = 000037            #5-161       7-926
LILLVE  = 000037            #5-156       7-1262
LIVDTE  = 000032            #5-182       7-517
LIVLII  = 000042            #5-169       7-1248
LLNAMT  = 000031            #5-149
LMARKE  = 000002            #5-141       6-231
LMSTRI  = 000052            #5-176
LMSUNU  = 000045            #5-175
LMXCN6  = 000050            #5-159       7-734
LMXLN6  = 000042            #5-150       7-746
LNAMET  = 000024            #5-181       7-268      7-285       7-564       7-859       7-1116
LNAMTL    000226 R          #5-149       5-149
LNSLIN  = 000031            #5-177       7-1209     7-1245      7-1280
LNSTER  = 000035            #5-174       7-913
LOCOPT  = 000042            #5-167       7-413
LONAMT  = 000042            #5-164       7-842
LOTSKT  = 000040            #5-168       7-809
LO.ADD    000022            #5-61       *7-1030    *7-1078     *7-1079
LO.ADR    000052            #5-61
LO.CLS    000002            #5-61       *7-346
LO.CON    000006            #5-61       *7-732
LO.COP    000010            #5-61       *7-407     *7-411      *7-820
LO.COS    000020            #5-61       *7-422
LO.CSR    000010            #5-61        5-61      *7-436
LO.CTL    000032            #5-61       *7-454
LO.DES    000002            #5-61        5-61       7-263
LO.DTE    000047            #5-61        7-506
LO.EVT    000004            #5-61        7-309
LO.GRO    000024            #5-61        7-559
LO.HOS    000040            #5-61       *7-1017    *7-1018
LO.HTM    000032            #5-61       *7-572
LO.INA    000062            #5-61       *7-709
LO.INC    000060            #5-61       *7-687
LO.LDT    000046            #5-61       *7-513
LO.LID    000030            #5-61       *7-642
LO.LIN    000004            #5-61       *7-744
LO.LTM    000034            #5-61       *7-653
LO.MAC    000022            #5-61       *7-664
LO.MDE    000024            #5-61       *7-676
LO.NAM    000044            #5-61        7-1096

                                                                 M  1
```

```
                                        .SBTTL  ERROR MESSAGES
                                ;****
                                ; ERROR MESSAGES
                                ;***

000000                                  ERMSG$  <Invalid line option>
000023                                  ERBLK$  LNER1
000026                                  ERMSG$  <Line not in system>
000050                                  ERBLK$  LNER2
000054                                  ERMSG$  <Circuit not in system>
000101                                  ERBLK$  CNER1
```

```
                                                        .SBTTL  ECHO - ECHO COMMANDS
                                               ;+
                                               ; **-ECHO-ECHO ACTIONS TAKEN AS A RESULT OF 'SET CEX ECHO'
                                               ;
                                               ;-

                                               ; MACRO TO MOVE ASCII

                                                        .MACRO MOVIT AA,BB
                                                        MOV     #AA,R0
                                                        MOV     #BB,R1
                                                        $$$=.
                                                        MOVB    (R1)+,(R0)+
                                                        BNE     $$$
                                                        .ENDM   MOVIT

 001250                                        ECHO:    CALL    $SAVAL              ; SAVE ALL REGISTERS
 001254  012746  001450'                                MOV     #100$,-(SP)         ; RETURN ADDRESS

 001260  122767  000002  000000G                        CMPB    #LS.PRO,LR.STS+RQB ; SET PROCESS ?
 001266  001007                                         BNE     3$                  ; NO
 001270                                                 MOVIT   SC,SET              ; MOVE IN SET
 001304  000412                                         BR      4$
 001306  122767  000005  000000G 3$:                    CMPB    #LX.PRO,LR.STS+RQB ; CLEAR PROCESS ?
 001314  001012                                         BNE     5$                  ; NO
 001316                                                 MOVIT   SC,CLEAR            ; MOVE IN CLEAR
 001332                          4$:                    ERRPT$  ECH2,$EROUT         ; WRITE OUT ECHO

 001342  122767  000003  000000G 5$:                    CMPB    #LS.LIN,LR.STS+RQB ; SET LINE ?
 001350  001020                                         BNE     8$                  ; NO
 001352  016700  000000G                                MOV     $$PRI,R0            ; CONVERT PRIORITY TO INTEGER
 001356                                                 ASR$    5,R0                ; ...
 001370  010067  000216                                 MOV     R0,PRI              ; SAVE CONVERTED PRIORITY
 001374                                                 MOVIT   SC,SET              ; SET
 001410  000412                                         BR      9$
 001412  122767  000006  000000G 8$:                    CMPB    #LX.LIN,LR.STS+RQB ; CLEAR LINE ?
 001420  001012                                         BNE     10$
 001422                                                 MOVIT   SC,CLEAR            ; CLEAR IN ASCII
 001436                          9$:                    ERRPT$  ECH3,$EROUT         ; ECHO

 001446                          10$:                   RETURN
 001450  000241                  100$:                  CLC                         ; NO ERRORS FROM ECHO MODE
 001452                                                 RETURN

                                               ; ECHO MESSAGE TEXT

 001454                                                 ERMSG$  <%A PROCESS %R>
 001471                                                 ERBLK$  ECH2,<SC,LR.PRO+RQB>
 001500                                                 ERMSG$  <%A LINE %R-%B-%B CSR %O VEC %O PRI %O>
 001545                                                 ERBLK$  ECH3,<SC,LR.LIN+RQB,LR.CTL+RQB,LR.UNT+RQB,$$CSR,$$VECT,PRI>

                                                        .NLIST  BEX
 001566                          ONOFF:                 .BLKB   4
 001572                          SC:                    .BLKB   6

 001600     123     105     124  SET:                   .ASCIZ  'SET'
```

```
                                                                  L  4
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45       PAGE 2
SYMBOL CROSS REFERENCE                                    CREF    04.00

SYMBOL   VALUE              REFERENCES

LO.NAM    000044            #5-69
LO.NET  = 000002            #5-69
LO.NNM    000022            #5-69
LO.OFL    000012            #5-69
LO.ONM    000004            #5-69
LO.OUT    000056            #5-69
LO.OWN    000026            #5-69
LO.PAR    000010            #5-69
LO.PRI    000012            #5-69
LO.RET    000064            #5-69
LO.RPA    000030            #5-69
LO.SCR    000016            #5-69
LO.SEG    000054            #5-69
LO.SKA    000014            #5-69
LO.SKN    000014            #5-69
LO.SNM    000035            #5-69
LO.TPA    000020            #5-69
LO.TRB    000034            #5-69
LO.TRI    000030            #5-69
LO.UNT    000033            #5-69
LO.VCT    000014            #5-69
LO.VER    000042            #5-69
LR.ACC    000020            #5-69
LR.ACO    000052            #5-69
LR.ADD    000012            #5-69
LR.ALI    000002            #5-69
LR.BLK    000010            #5-69
LR.CTL    000004            #5-69      10-229     *10-243     12-347     *12-361     15-544
LR.DES    000010            #5-69
LR.HLP    000002            #5-69
LR.LIN    000002            #5-69       5-69        5-69       5-69        5-69       5-69       5-69       10-222     *10-242
                              12-340   *12-360     15-544
LR.NAM    000012            #5-69
LR.PAS    000041            #5-69
LR.PRO    000002            #5-69      15-542
LR.STS    000000            #5-69      15-514      15-518     15-523      15-530
LR.TRI    000006            #5-69     *10-245     *12-363    *12-370     *12-371
LR.TYP    000002            #5-69
LR.UCB    000016            #5-69
LR.UIC    000020            #5-69
LR.UNT    000005            #5-69      10-236     *10-244     12-354     *12-362     15-544
LS.ALI  = 000016            #5-69
LS.CEX  = 000001            #5-69
LS.CIR  = 000033            #5-69      9-189
LS.CXO  = 010000            #5-69
LS.CX5  = 000400            #5-69
LS.ECH  = 001000            #5-69
LS.FDX  = 004000            #5-69
LS.HDX  = 010000            #5-69
LS.HLP  = 000012            #5-69
LS.LIN  = 000003            #5-69      9-187       15-523
LS.LMC  = 000007            #5-69
```

```
                                                                  M  4
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45       PAGE 3
```

```
VNSD       CREATED BY  MACRO  ON 29-JUN-85 AT 02:46     PAGE 2
MACRO CROSS REFERENCE                                   CREF   04.00

MACRO NAME       REFERENCES

DCBDF$           #5-56        5-60
LBLDF$           #5-56        5-66
PCBDF$           #5-56        5-85
SCBDF$           #5-56        5-73
UCBDF$           #5-56        5-79
```

```
**FILE**ID**VNTI
```

```
                                           SWTCHO                        ; RESET OUTPUT BUFFER DEFAULT     ;[MP01]
                                   145$:                                                                   ;[MP01]
                                           .ENDC                                                           ;[MP01]



                                   ;
                                   ; NON 11/70 MAPPED SYSTEM
                                   ;
001014  010267  000000G            150$:   MOV     R2,$QBIAS             ; SETUP FREE CORE POINTER
001020  010267  000000G                    MOV     R2,$QSTRT             ; SAVE ORIGINAL $QBIAS VALUE
001024  010267  000000G                    MOV     R2,$RBLCK             ; MAP TO FIRST FREE SEGMENT
001030  005067  000000G                    CLR     $BFR                  ; NO NEXT SEGMENT
001034  010367  000002G                    MOV     R3,$BFR+2             ; LENGTH OF THIS SEGMENT
001040                                     PUTAD   #0,#4                 ; INITIALIZE POOL
                                   ;
                                   ; DO DECNET CONFIGURATION FILE SCAN
                                   ;
001060                                     CALL    $CFG                  ; SCAN CETAB FOR DECNET
001064  103004                             BCC     200$                  ; OK

001066                             190$:   NTIERR  406                   ; CONFIG FILE SCAN FAILURE (DUMP IMAGE)

001076                             200$:   GETRV   $DECPT,#D$END         ; READ DECNET HOME BLOCK
001116  016767  000102' 000054G            MOV     NBRA,D$NBRA+$BFR      ; STORE NI ROUTER ADJACENCIES
001124                                     PUTRC
001130  005727  000000G                    TST     #$NTCFP               ; PSI SCANNER LINKED?
001134  001403                             BEQ     210$                  ; NO, BETTER NOT CALL IT!
001136                                     CALL    $NTCFP                ; SCAN CETAB FOR PSI COMPONENTS
001142  103751                             BCS     190$                  ; FAILED, GIVE ERROR

001144  005727  000000G            210$:   TST     #$NTCFS               ; SNA SCANNER LINKED?
001150  001403                             BEQ     390$                  ; NO, DON'T CALL IT!
001152                                     CALL    $NTCFS                ; SCAN CETAB FOR SNA COMPONENTS
001156  103743                             BCS     190$                  ; FAILED, GIVE ERROR
                                   ;
                                   ; ALLOCATE BUFFERS
                                   ;
001160                             390$:   CALL    $CCBS                 ; ALLOCATE CCBS
001164  103004                             BCC     395$                  ; NO ERRORS
001166                                     NTIERR  7                     ; RESOURCE ERROR ALLOCATING CCBS

001176  005067  000000C            395$:   CLR     $NTLHB+.SBPCB         ; ASSUME NOT SERVER BASE
001202  005767  000000G                    TST     SBFLG                 ; SERVER BASE SYSTEM ?
001206  001421                             BEQ     400$                  ; IF NOT - BRANCH
001210  022767  064000  000002G            CMP     #64000,NTPSIZ+2       ; CHECK AGAINST 104KW LIMIT
001216  003411                             BLE     398$                  ; IF RDB/SDB SIZE TOO LARGE - BRANCH

001220  005267  000000G                    INC     SBPOOL                ; SET SERVER BASE FLAG
001224                                     CALL    $CRPAR                ; CREATE SERVER BASE POOL PARTITION

001230  010167  000000G                    MOV     R1,SBPCB              ; SAVE SBPOOL PCB
001234  010167  000000C                    MOV     R1,$NTLHB+.SBPCB      ; USED FOR DEAL.
001240  001015                             BNE     410$                  ; IF PCB ALLOCATION SUCCESSFUL - BRANCH
001242                             398$:   NTIERR  2                     ; REPORT ERROE


```

```
                                                .SBTTL  DYNAMICALLY ALLOCATE SUB-PARTITION

                                        ;+
                                        ;       **-$CRPAR
                                        ;
                                        ;       THIS ROUTINE IS CALLED TO DYNAMICALLY CREATE A NETWORK
                                        ;       BUFFER POOL BY ALLOCATING A SUB-PARTITION IN ANY OF THE
                                        ;       SYSTEM CONTROLLED PARTITIONS FOUND IN THE PARTION LIST.
                                        ;       ALL ERROR REPORTING IS EXTERNAL TO THIS SUBROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NTPSIZ  - NUMBER OF 32-WORD BLOCKS IN NTPOOL
                                        ;       NTPNAM  - RAD50 NAME OF NTPOOL
                                        ; OUTPUTS:
                                        ;       R1      - PCB ADDRESS IF SUB-PARTITION ALLOCATED
                                        ;
                                        ;-

 003366  005067  000000G                $CRPAR::CLR     TOPFLG                  ; TURN OFF TOP LOADING
 003372  016767  000000G 000000G                MOV     $PARHD,$BFR             ; ADDRESS OF PARTITION LISTHEAD
 003400  016767  000000G 000000G                MOV     .PLGTH,$RSIZE           ; LENGTH OF TRANSFERS

 003406  005767  000000G                        TST     SBPOOL                  ; SERVER BASE SYSTEM ?
 003412  001006                                 BNE     10$                     ; IF YES - BRANCH

 003414  005767  000000G                        TST     NTPPAR                  ; WAS A PARTITION NAME INCLUDED?
 003420  001003                                 BNE     10$                     ; IF NE, YES
 003422  012767  026226  000000G                MOV     #^RGEN,NTPPAR           ; ELSE DEFAULT TO "GEN"

 003430  016701  000000G                10$:    MOV     $BFR,R1                 ; GET NEXT PCB ADDRESS
 003434  001475                                 BEQ     30$                     ; FAILED TO ALLOCATE POOL
 003436                                         GETRV   R1                      ; GET IMAGE INTO BUFFER

                                                .IF     NDF,R$$MPL
 003446  032767  000040  000030G                BIT     #PS.SYS,P.STAT+$BFR ; SYSTEM CONTROLLED PARTITION ?
 003454  001765                                 BEQ     10$                     ; NO, KEEP LOOKING
                                                .ENDC

 003456  005767  000000G                        TST     SBPOOL                  ; SERVER BASE SYSTEM ?
 003462  001411                                 BEQ     15$                     ; IF NOT - BRANCH
 003464  026767  000004G 000000G                CMP     P.NAM+$BFR,SBPAR  ; CHECK FOR PARTITION
 003472  001356                                 BNE     10$                     ; IF NO MATCH - BRANCH
 003474  026767  000006G 000002G                CMP     P.NAM+2+$BFR,SBPAR+2 ; CHECK FOR PARTITION
 003502  001352                                 BNE     10$                     ; IF NO MATCH - BRANCH
 003504  000446                                 BR      20$                     ; PARTITION FOUND - BRANCH


 003506  026767  000004G 000000G 15$:           CMP     P.NAM+$BFR,NTPPAR ; RIGHT PARTITION ?
 003514  001345                                 BNE     10$                     ; NO
 003516  026767  000006G 000002G                CMP     P.NAM+2+$BFR,NTPPAR+2 ; MATCH EVERYTHING
 003524  001341                                 BNE     10$                     ; MISMATCH
 003526  005767  000000G                        TST     TOPFLG                  ; TOP-DOWN LOADING REQUESTED ?
 003532  001433                                 BEQ     20$                     ; IF EQ, NO .. OKAY


                                        ;
                                        ; SEE IF TOP-DOWN LOADING IS POSSIBLE
                                        ;
                                        .SBTTL  ALLOCATE SUB-PARTITION OUT OF MAIN PARTITION

                                ;+
                                ;       CRSUB
                                ;
                                ;       THIS ROUTINE IS CALLED EACH TIME A SYSTEM CONTROLLED PARTITION
                                ;       IS FOUND. AN ATTEMPT IS MADE TO ALLOCATE NTPOOL OUT OF THE
                                ;       PARTITION UNTIL A LARGE ENOUGH HOLE IS FOUND OR UNTIL ALL PAR-
                                ;       TITIONS HAVE BEEN SCANNED AND IT IS DETERMINED THAT NO PARTITION
                                ;       HAS A HOLE LARGE ENOUGH FOR NTPOOL.
                                ;
                                ; INPUTS:
                                ;       R5      - MAIN PARTITION PCB ADDRESS
                                ;       NTPSIZ  - NUMBER OF 32-WORD BLOCKS IN NTPOOL
                                ;       NTPNAM  - RAD50 NAME OF NTPOOL
                                ;
                                ; OUTPUTS:
                                ;       C-BIT SUCCESS/FAILURE
                                ;       R0,R1,R2,R3,R4 - DESTROYED
                                ;       R5      - SUB-PARTITION PCB ADDRESS IF SUCCESSFUL
                                ;-

004640  016701  000000G         CRSUB:  MOV     .PLGTH,R1       ; GET PCB LENGTH
004644                                  CALL    $ALL15          ; TRY TO ALLOCATE SUB-PARTITION PCB
004650  103002                          BCC     3$              ; BR IF SUCCESS
004652  000167  000526                  JMP     102$            ; IF CS, FAILURE
004656  012701  000000G         3$:     MOV     #$BFR,R1        ; GET PCB ADDRESS
004662  016702  000000G                 MOV     .PLGTH,R2       ;  AND LENGTH
004666  006202                          ASR     R2              ; CONVERT TO A WORD COUNT
004670  005021                  5$:     CLR     (R1)+           ; ZERO THE ENTIRE SUB-PCB
004672  005302                          DEC     R2
004674  001375                          BNE     5$              ; ...

004676  005767  000000G                 TST     SBPOOL          ; SERVER BASE SYSTEM ?
004702  001421                          BEQ     7$              ; IF NOT BRANCH
004704  016767  000000G 000000G         MOV     NTPSIZ,NTPSAV   ; SAVE POOL.. SIZE
004712  016767  000002G 000000G         MOV     NTPSIZ+2,NTPSIZ ; SET UP SERVER BASE BUFFER POOL SIZE
004720  016767  000000G 000004G         MOV     SBNAM,P.NAM+$BFR ; SET UP PCB NAME
004726  016767  000002G 000006G         MOV     SBNAM+2,P.NAM+2+$BFR ; ...

004734  016746  000000G                 MOV     PBLOX,-(SP)     ; SAVE NETWORK POOL.. SIZE ALLOCATED
004740  016746  000000G                 MOV     PSTART,-(SP)    ; SAVE STARTING ADDRESS (JUST IN CASE)
004744  000406                          BR      9$              ; JOIN COMMON CODE

004746  016767  000000G 000004G 7$:     MOV     NTPNAM,P.NAM+$BFR ; SET NAME
004754  016767  000002G 000006G         MOV     NTPNAM+2,P.NAM+2+$BFR ;*          ...
004762  010567  000012G         9$:     MOV     R5,P.MAIN+$BFR  ; MAIN PCB ADDRESS
004766  016767  000000G 000016G         MOV     NTPSIZ,P.BLKS+$BFR ; SIZE IN 32. WORD BLOCKS
004774  016767  000000G 000000G         MOV     NTPSIZ,PBLOX    ; SAVE IT FOR LATER

                                        .IF     DF,R$$MPL
                                        MOV     #PS.COM!PS.NSF!PS.FXD!PS.CHK,P.STAT+$BFR ; COMMON, ETC.
                                        MOV     #P2.RON,P.ST2+$BFR                       ; SO NO TASK ATTACHES
                                        .IFF
005002  012767  000640  000030G         MOV     #PS.COM!PS.NSF!PS.SYS,P.STAT+$BFR ; COMMON, NO SHUFFLING
                                        .ENDC

                                        .SBTTL  SLT$DF PARSE TABLES
                               ;+
                               ;        STATE TRANSITIONS FOR THE 'SLT$DF' MACRO
                               ;-

006324                                  STATE$  SLTDF
006324                                   TRAN$   $RAD50,,.PRONM
006324                                  STATE$
006324                                   TRAN$   <',>
006324                                  STATE$
006324                                   TRAN$   !PRONAM
006324                                  STATE$
006324                                   TRAN$   !PRONAM
006324                                  STATE$  SLTFLG
006324                                   TRAN$   %LF.MFL%,SLTFLG,,LF.MFL,MRKFL
006324                                   TRAN$   %LF.ENA%,SLTFLG,,LF.ENA,MRKFL
006324                                   TRAN$   %LF.SER%,SLTFLG,,LF.SER,MRKFL
006324                                   TRAN$   '!,SLTFLG
006324                                   TRAN$   $RAD50,SLTFLG
006324                                   TRAN$   <',>
006324                                  STATE$
006324                                   TRAN$   $NUMBR,,.SCTL
006324                                  STATE$
006324                                   TRAN$   <',>
006324                                  STATE$
006324                                   TRAN$   $NUMBR,,.SUNT
006324                                  STATE$
006324                                   TRAN$   !END,$EXIT
006324                                   TRAN$   <',>
006324                                  STATE$
006324                                   TRAN$   <',>,SCTIM
006324                                   TRAN$   $STRNG
006324                                  STATE$
006324                                   TRAN$   !END,$EXIT
006324                                   TRAN$   <',>
006324                                  STATE$  SCTIM
006324                                   TRAN$   $NUMBR,SCHELO,.SCTIM
006324                                   TRAN$  <',>,SCHEL
006324                                  STATE$  SCHELO
006324                                   TRAN$  !END,$EXIT
006324                                   TRAN$  <',>
006324                                  STATE$  SCHEL
006324                                   TRAN$  !END,$EXIT
006324                                   TRAN$  $NUMBR,BRADJ
006324                                   TRAN$  <',>
006324                                  STATE$  BRADJ
006324                                   TRAN$  <',>,BRADJ1
006324                                   TRAN$  !END,$EXIT
006324                                  STATE$  BRADJ1
006324                                   TRAN$  $NUMBR,$EXIT,.SBRA


```

```
   2372
   2373 006324                                          STATE$  FILSPC
   2374 006324                                           TRAN$  <'[>,UIC
   2375 006324                                           TRAN$  $RAD50,TOK,GETSTR
   2376 006324                                           TRAN$  $LAMDA
   2377
   2378 006324                                          STATE$  UIC
   2379 006324                                           TRAN$  $STRNG,,GETSTR
   2380
   2381 006324                                          STATE$
   2382 006324                                           TRAN$  <',>,FILSPC,GRPUIC
   2383 006324                                           TRAN$  <']>,FILSPC,USRUI1
   2384 006324                                          STATE$ TOK
   2385 006324                                           TRAN$  <':>,FILSPC,DEV
   2386 006324                                           TRAN$  <']>,FILSPC,USRUIC
   2387 006324                                           TRAN$  <'.>,EXTEN,FILES
   2388 006324                                           TRAN$  $LAMDA,$EXIT,FILES
   2389
   2390 006324                                          STATE$  EXTEN
   2391 006324                                           TRAN$  $STRNG,,GETSTR
   2392
   2393 006324                                          STATE$
   2394 006324                                           TRAN$  $LAMDA,$EXIT,EXT
   2395
```

```
                                                        .SBTTL  ACTION ROUTINE: STA$DF MACRO SEEN
                                                ;+
                                                ;       **-.STADF-STA$DF MACRO SEEN
                                                ;
                                                ;       A DEFINTION OF ANOTHER STATION LINE TABLE ENTRY HAS BEEN SEEN.
                                                ;
                                                ;-
                                                ;
     006576  005267  000036'                    .STADF: INC     STAN                    ; UPDATE STATION NUMBER
     006602                                             RETURN
```

```
                                             .SBTTL  ACTION ROUTINE: REMOTE ADDRESS FOUND
                                    ;+
                                    ; *** - .RNUM - REMOTE NODE ADDRESS FOUND
                                    ;
                                    ; INPUTS:
                                    ;        .PNUMB = THE REMOTE NODE ADDRESS
                                    ;
                                    ; OUTPUTS:
                                    ;        A REMOTE NODE BLOCK IS ALLOCATED AND FILLED IN
                                    ;
                                    ;-

011330  005767  000000G             .RNUM:   TST     .PNUMB            ; NODE ADDRESS IN RANGE ?
011334  001524                               BEQ     111$              ; IF EQ, NO
011336  012701  000014                       MOV     #R.LEN,R1         ; GET THE LENGTH OF A REMOTE BLOCK
011342                                       CALL    $ALPOL            ; TRY TO ALLOCATE A REMOTE BLOCK
011346  103513                               BCS     101$              ; IF CS, COULDN'T

011350  012701  000000G                      MOV     #$BFR,R1          ; POINT AT I/O BUFFER
011354  016761  000030' 000002               MOV     NAME,R.NAM(R1)    ; STORE THE NODE NAME
011362  016761  000032' 000004               MOV     NAME+2,R.NAM+2(R1) ;     ...
011370  016761  000034' 000006               MOV     NAME+4,R.NAM+4(R1) ;     ...
011376  016761  000100' 000010               MOV     REMARE,R.ADD(R1)  ; STORE AREA ADDRESS
011404  056761  000000G 000010               BIS     .PNUMB,R.ADD(R1)  ; STORE THE NODE ADDRESS
011412  005061  000012                       CLR     R.FLAG(R1)        ; INITIALIZE FLAGS WORD
011416  010067  000004'                      MOV     R0,CURUNM         ; SAVE UNMAPPED ADDRESS
011422                                       CEACC$  R0                ; CONVERT TO MAPPED ADDRESS
011432                              10$:     PUTAD   R0,#R.LEN         ; WRITE OUT THE REMOTE BLOCK
011450  016701  000010G                      MOV     $BFR+R.ADD,R1     ; GET REMOTE NODE ADDRESS
011454  016700  000004'                      MOV     CURUNM,R0         ; RETRIEVE UNMAPPED ADDRESS
011460                                       CALL    QUICK             ; ATTEMPT FAST LIST PROCESSING
011464  103003                               BCC     20$               ; IF SUCCESS - BRANCH
011466  000241                               CLC                       ; IF FAILURE - USE STANDARD SEARCH
011470                                       CALL    FINREM            ; FIND WHERE REMOTE GOES IN LIST

011474                              20$:     SWTCHI  #$HMBUF           ; USE TEMPORARY BUFFER
011502                                       GETRV   $DECPT,#D$END     ; READ DECNET HOME BLOCK
011522                                       SWTCHI                    ; RESET TO DEFAULT BUFFER

011530  012702  000004G                      MOV     #$HMBUF+D$RNN+2,R2 ; POINT TO LISTHEAD
011534                                       CALL    QINS              ; INSERT ENTRY INTO QUEUE

011540                                       SWTCHO  #$HMBUF           ; USE TEMPORARY BUFFER
011546                                       PUTRC   $DECPT,#D$END     ; WRITE BACK DECNET HOME BLOCK
011566                                       SWTCHO                    ; RESET DEFAULT BUFFER

011574                                       RETURN                    ; RETURN TO USER MODE AND TO CALLER
                                    ;
                                    ; ERROR CONDITIONS
                                    ;
011576                              101$:    NTIERR  23                ; FAILED TO ALLOCATE A REMOTE BLOCK
011606  062716  000002              111$:    ADD     #2,(SP)           ; REJECT TRANSITION
011612                                       RETURN

                                    ;+
                                    ; *** - .RARE - REMOTE AREA ADDRESS
                                    ;-
```

```
                                                 .SBTTL  ACTION ROUTINE: SET LOGGING STATE

                                         ;+
                                         ;
                                         ; L.LGST - SET LOGGING STATE
                                         ;
                                         ; INPUTS:
                                         ;        .PNUMB - LOGGING STATE
                                         ;
                                         ;-
 013206  105767  000001G                 L.LGST: TSTB    .PNUMB+1                ; MUST BE BYTE VALUE
 013212  001010                                  BNE     10$                     ; BR IF NOT
 013214  126727  000000G 000007                  CMPB    .PNUMB,#7               ; LEGAL VALUES ARE 0-7
 013222  101004                                  BHI     10$                     ; BR IF ILLEGAL VALUE
 013224  116767  000000G 000000G                 MOVB    .PNUMB,$LGSTT           ; STORE LOGGING STATE
 013232  000402                                  BR      20$                     ; AND EXIT
 013234  062716  000002                  10$:    ADD     #2,(SP)                 ; REJECT THE TRANSITION
 013240                                  20$:    RETURN
```

```

                                ;+
                                ; HRDADD - STORE HARDWARE ADDRESS
                                ;
                                ; INPUTS -
                                ;         .PSTPT - PARSED STRING ADDRESS
                                ;         .PSTCN - STRING SIZE
                                ;
                                ; OUTPUTS -
                                ;         HARDWARE ADDRESS CONVERTED
                                ;-

014550  010346                  HRDADD::MOV     R3,-(SP)        ; SAVE REG
014552  016700  000000G                 MOV     .PSTPT,R0       ; GET PARSED STRING ADDRESS
014556  012702  000140'                 MOV     #SERADD,R2      ; GET HARDWARE ADDRESS BUFFER
014562                          5$:     CALL    HEXIND          ; CONVERT HEX

014566  006303                          ASL     R3              ; POSITION DIGIT
014570  006303                          ASL     R3              ; ...
014572  006303                          ASL     R3              ; ...
014574  006303                          ASL     R3              ; ...

014576  110312                          MOVB    R3,(R2)         ; STORE DIGIT
014600                                  CALL    HEXIND          ; CONVERT HEX
014604  150322                          BISB    R3,(R2)+        ; STORE DIGIT

014606  022702  000146'                 CMP     #SERTMP,R2      ; END OF BUFFER ?
014612  001363                          BNE     5$
014614  012603                          MOV     (SP)+,R3        ; RESTORE REG

014616                                  RETURN



014620                          SEREND::CALL    SERALL          ; ALLOCATE SERVICE BLOCK
014624  103401                          BCS     101$            ; IF ALLOCATION FAILURE BRANCH

014626                                  RETURN

014630                          101$:   NTIERR  33              ; SERVICE BLOCK ALLOCATION FAILURE

```

```
  4473
```

Symbol table

```
ERR4     000522R   002  E11LEN=  000046         FE.LSI=  000400         FS.RNG=  011000         F3.NWK=  000002
ERR40    002264R   002  E12LEN=  000046         FE.MUP=  000002         FS.RST=  000000         F3.PMN=  004000
ERR41    002320R   002  E13LEN=  000035         FE.MXT=  040000         FS.RTN=  001000         F3.PRO=  000040
ERR42    002364R   002  E14LEN=  000042         FE.NLG=  100000         FS.SET=  005000         F3.RLK=  020000
ERR43    002420R   002  E15LEN=  000034         FE.OFF=  001000         FS.SFC=  005000         F3.SHF=  040000
ERR44    002454R   002  E16LEN=  000034         FE.PKT=  000100         FS.SFR=  006000         F3.STM=  000010
ERR45    002512R   002  E17LEN=  000030         FE.PLA=  000020         FS.SFS=  004000         F3.TCM=  002000
ERR5     000566R   002  E2LEN =  000062         FE.X25=  004000         FS.SPW=  040000         F3.UDS=  000020
ERR6     000622R   002  E20LEN=  000034         FF.ADD=  000020         FS.STM=  000000         F3.WAT=  010000
ERR7     000674R   002  E21LEN=  000056         FF.CIR=  000040         FS.STP=  002000         F3.XHR=  000100
ER$ABM=  000010         E22LEN=  000040         FF.CON=  000001         FS.STR=  001000         F3.11S=  000400
ER$ABO=  000046         E23LEN=  000055         FF.FIL=  000002         FS.TRM=  003000         G        002432R   003
ER$ABT=  000011         E24LEN=  000034         FF.HST=  040000         FS.WLB=  001000         GETSTR   016760RG
ER$ACC=  000042         E25LEN=  000031         FF.LIN=  000010         FS.XKL=  002000         GRPUIC   017062RG
ER$CDI=  000052         E26LEN=  000022         FF.MOD=  000100         FS.XOF=  010000         G$$TPP=  000000
ER$COM=  000047         E27LEN=  000033         FF.MON=  000004         FS.XON=  007000         G$$TSS=  000000
ER$FMT=  000005         E3LEN =  000050         FF.MSK=  000077         FS.ZER=  002000         G$$TTK=  000000
ER$MLB=  000006         E30LEN=  000012         FF.PRT=  000200         F$$LVL=  000001         G$$WRD=  000000
ER$NNF=  000012         E31LEN=  000034         FF.QUL=  000370         F.ADD    000016         H        002456R   003
ER$NOD=  000002         E32LEN=  000041         FF.REM=  100000         F.CEV    000030         HEXIND   015330R
ER$NSL=  000013         E33LEN=  000037         FIELDC=  000003         F.CIR    000024         HEXSTR   000164RG  002
ER$NSR=  000003         E34LEN=  000037         FILBUF   000222RG  002  F.CLS    000002         HF.CIS=  000200
ER$RES=  000001         E35LEN=  000036         FILDF    001754R   003  F.EVT    000004         HF.EIS=  000002
ER$STA=  000051         E36LEN=  000032         FILES    014142RG       F.FLG    000014         HF.FPP=  100000
ER$UOB=  000004         E37LEN=  000032         FILFLG   000252RG  002  F.HIBK=  ****** GX      HF.QB =  000004
EVENT    000046R   002  E4LEN =  000044         FILSPC   002072R   003  F.LEN    000016         HF.UBM=  000001
EVTDF    001550R   003  E40LEN=  000034         FINFIL   013104R        F.LIN    000020         HRDADD   014550RG
EVTMSK   001610R   003  E41LEN=  000044         FINOBJ   013632R        F.LNK    000000         HSAD     014404RG
EXT      014202RG       E42LEN=  000034         FINREM   013732R        F.MOD    000022         HSAR     014364RG
EXTEN    002156R   003  E43LEN=  000034         FLAG     000062R   002  F.NRBD=  ****** GX      I        002502R   003
EXTN  =  000003         E44LEN=  000036         FLSZ  =  000020 G       F.NXT    000000         IBUF     000044RG  003
E$NBR    000014         E45LEN=  000037         FLTFLG   000044R   002  F.REM    000026         IBUF0    000000RG  003
E$NBS    000020         E5LEN =  000034         FNBEXT=  000006         F.RSIZ=  ****** GX      IBUF2    000124RG  003
E$NCR    000034         E6LEN =  000051         FNBLN    000264RG  002  F.SEV    000004         IBUF3    000254RG  003
E$NCS    000036         E7LEN =  000046         FNBSZ =  000044 G       F.URBD=  ****** GX      IBUF4    000306RG  003
E$NIC    000044         F        002406R   003  FNBUFF   000266RG  002  F.USE    000002         IBUF5    000324RG  003
E$NLEN   000050         FC.CCP=  000020         FPDV     015402R        F.VAR    000003         IE.EOF=  ****** GX
E$NLLA   000012         FC.CTL=  000006         FSLT     015450R        F2.ACN=  000020         INAC     010624R
E$NLNK   000000         FC.KCP=  000016         FS.AST=  000000         F2.AHR=  010000         INCT     010500R
E$NML    000040         FC.KIL=  000004         FS.CIB=  002000         F2.DAS=  000001         INIBUF   016002R
E$NMR    000024         FC.MAN=  000024         FS.CRA=  001000         F2.DPR=  000400         IN.DAT=  000400
E$NMS    000030         FC.MLD=  000026         FS.DIS=  013000         F2.EVT=  000010         IN.ILS=  000001
E$NNOD   000002         FC.PCT=  000030         FS.DVC=  001000         F2.GGF=  002000         IO.WVB=  ****** GX
E$NRT    000042         FC.PWR=  000022         FS.ENB=  012000         F2.IRR=  001000         IPL      011022R
E$NRTP   000005         FC.RCE=  000002         FS.EXI=  001000         F2.LIB=  000002         ISTATE   000466RG  003
E$NSEG   000010         FC.RCP=  000014         FS.GET=  006000         F2.MP =  000004         I$$RAR=  000000
E$NTIM   000046         FC.TIM=  000010         FS.HLT=  000000         F2.POL=  000100         I$$RDN=  000000
E$NUSE   000004         FC.XCP=  000012         FS.INI=  000000         F2.RAS=  004000         KBUF     000002RG  004
E$STRT   000006         FC.XME=  000000         FS.KIL=  000000         F2.RBN=  020000         KBUF0    000000RG  004
E$$XPR=  000000         FE.CAL=  000040         FS.LCL=  100000         F2.SDW=  000040         KBUF2    000004RG  004
E.CON    012676R        FE.CEX=  020000         FS.LTM=  001000         F2.STP=  100000         KBUF3    000014RG  004
E.FIL    012706R        FE.DRV=  000010         FS.MNT=  004000         F2.SWP=  040000         KBUF4    000020RG  004
E.MON    012716R        FE.DYM=  010000         FS.MSN=  014000         F2.WND=  000200         KBUF5    000022RG  004
E.MSK    012634R        FE.EXP=  000200         FS.REA=  001000         F3.AST=  000200         KDSAR0=  172360
E.NUM    012554R        FE.EXT=  000001         FS.RET=  000000         F3.CLI=  001000         KDSDR0=  172320
E1LEN =  000031         FE.EXV=  000004         FS.REZ=  003000         F3.CRA=  000001         KINAR0=  172340
E10LEN=  000040         FE.FDT=  002000         FS.RLB=  002000         F3.EIS=  000004         KINAR5=  172352
```

```
                                                                    M  1
VNCP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:44        PAGE 4
SYMBOL CROSS REFERENCE                                      CREF   04.00
SYMBOL   VALUE             REFERENCES

LO.NET  = 000002           #5-61        7-275      7-279
LO.NNM    000022           #5-61        7-1028
LO.OFL    000012           #5-61       *7-821     *7-831
LO.ONM    000004           #5-61       *7-806     *7-807     *7-822     *7-823
LO.OUT    000056           #5-61       *7-698
LO.OWN    000026           #5-61       *7-840
LO.PAR    000010           #5-61       *7-1124    *7-1125
LO.PRI    000012           #5-61       *7-878
LO.RET    000064           #5-61       *7-720
LO.RPA    000030           #5-61        7-1132
LO.SCR    000016           #5-61       *7-924
LO.SEG    000054           #5-61       *7-938
LO.SKA    000014           #5-61       *7-1050    *7-1054
LO.SKN    000014           #5-61        7-1048
LO.SNM    000035           #5-61        7-1159
LO.TPA    000020           #5-61        7-1169
LO.TRB    000034           #5-61
LO.TRI    000030           #5-61       *7-1201    *7-1219
LO.UNT    000033           #5-61       *7-1237
LO.VCT    000014           #5-61       *7-1260
LO.VER    000042           #5-61
LPASWT  = 000030           #5-158       7-1148     7-1185
LPNAMT  = 000034           #5-148       7-869
LPRNMT  = 000036           #5-151       7-1127
LPSWFT  = 000036           #5-173       7-490
LR.ACC    000020           #5-61
LR.ACO    000052           #5-61        7-466
LR.ADD    000012           #5-61       *7-551     *7-994
LR.ALI    000002           #5-61        7-975
LR.BLK    000010           #5-61
LR.CTL    000004           #5-61       *7-452
LR.DES    000010           #5-61       *7-965     *7-966      7-983
LR.HLP    000002           #5-61       *10-1368
LR.LIN    000002           #5-61        5-61       5-61       5-61       5-61       5-61       5-61      *7-640      7-850
                             11-1404
LR.NAM    000012           #5-61        7-992
LR.PAS    000041           #5-61        7-483
LR.PRO    000002           #5-61       *7-867
LR.STS    000000           #5-61        12-1428    12-1430
LR.TRI    000006           #5-61       *7-1199
LR.TYP    000002           #5-61       *7-819
LR.UCB    000016           #5-61       *7-911
LR.UIC    000020           #5-61        7-536
LR.UNT    000005           #5-61       *7-1235
LSYNER  = 000023           #5-146       6-243
LS.ALI  = 000016           #5-61        7-627
LS.CEX  = 000001           #5-61
LS.CIR  = 000033           #5-61
LS.CXO  = 010000           #5-61
LS.CX5  = 000400           #5-61
LS.ECH  = 001000           #5-61
LS.FDX  = 004000           #5-61
```

```
                                                                    N  1
```

```
                                .SBTTL  $VNP - VIRTUAL NCP ENTRY POINT
                        ;+
                        ; **-$VNP-VIRTUAL NCP ENTRY POINT
                        ;
                        ;       THIS IS THE ENTRY POINT FOR THE VIRTUAL NETWORK CONTROL
                        ;       PROCESSOR. MOST NETWORK FUNCTIONS THAT MODIFY MEMORY ARE
                        ;       INCLUDED IN THIS PROGRAM SO THAT THE IDENTICAL OPERATION
                        ;       CAN BE DONE TO A SYSTEM IMAGE. THIS IMAGE COULD THEN BE
                        ;       DOWN-LINE LOADED, OR BOOTED.
                        ;
                        ;       A PARTIAL LISTING OF LEGAL COMMANDS:
                        ;
                        ;       VNP> SET SYSTEM [ECHO]
                        ;       VNP> SET EXECUTOR STATE ON [UNFIX]
                        ;                                  [FIXED]
                        ;                               OFF
                        ;                         RECEIVE [PASSWORD] <PASSWORD>
                        ;                         TRANSMIT [PASSWORD] <PASSWORD>
                        ;                         HOST <NODEADDRESS>
                        ;                         VERIFICATION [STATE] ON
                        ;                                              OFF
                        ;       VNP> SET LINE <LINEID> ALL
                        ;                              STATE ON
                        ;                                    OFF
                        ;                              [CONTROLLER] CSR X
                        ;                              VECTOR X
                        ;                              PRIORITY X
                        ;                              DUPLEX HALF/FULL
                        ;                              [LOCATION] TOP
                        ;       VNP> SET PROCESS <NAME> ALL
                        ;                               PARTITION X
                        ;                               MAXIMUM CONTROLLERS X
                        ;                               MAXIMUM LINES X
                        ;       VNP> SET NODE X NAME <NODENAME>
                        ;                       LINE <LINEID>
                        ;
                        ;       VNP> CLEAR SYSTEM [ALL]
                        ;       VNP> CLEAR PROCESS <NAME> [ALL]
                        ;       VNP> CLEAR LINEID <LINE> ALL
                        ;       VNP> CLEAR EXECUTOR HOST
                        ;                           RECEIVE PASSWORD
                        ;                           TRANSMIT PASSWORD
                        ;       VNP> CLEAR NODE X NAME
                        ;                         LINE
                        ;
                        ; WHERE:
                        ;       <NAME>          - XPT           ( ANY LEGAL PROCESS NAME )
                        ;       <LINEID>        - DMC 0 0
                        ;       <NODENAME>      - MANILA        ( 1-6 ALPHA )
                        ;       <PASSWORD>      - SHAZZAM       ( 1-8 ALPHA )
                        ;-

000104                  $VNP::
000104                          CALL    $VINI           ; ONCE ONLY INITIALIZATION CODE
000110  103407                  BCS     $PANIC          ; EXIT WITHOUT A MURMUR
000112  000241          10$:    CLC                     ; CLEAR C-BIT
000114                          CALL    $VNCP           ; PARSE COMMAND
```

```
    551 001604     103      114      105  CLEAR:  .ASCIZ  'CLEAR'
    552                                           .EVEN
    553 001612  000000                    PRI:    .WORD   0               ; PRIORITY AS INTEGER
    554                                           .LIST   BEX
```

```
                                                                  M 4
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45     PAGE 3
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE                REFERENCES

LS.LOG  = 000020             #5-69
LS.MOD  = 000036             #5-69
LS.NOD  = 000014             #5-69
LS.NTI  = 000013             #5-69
LS.OBJ  = 000022             #5-69
LS.OPT  = 000400             #5-69
LS.PRO  = 000002             #5-69        15-514
LS.TOP  = 002000             #5-69
LS.TST  = 000010             #5-69
LS.UNF  = 020000             #5-69
LS.XIT  = 000011             #5-69
LX.ALI  = 000017             #5-69
LX.CEX  = 000004             #5-69
LX.CIR  = 000034             #5-69
LX.LIN  = 000006             #5-69        9-188        15-530
LX.LOG  = 000021             #5-69
LX.NOD  = 000015             #5-69
LX.OBJ  = 000023             #5-69
LX.PRO  = 000005             #5-69        15-518
L.CTL   = ******  GX          10-229      10-243       12-347       12-361
L.DDM   = ******  GX          14-488
L.FLG   = ******  GX          17-664
L.NSTA  = ******  GX          12-367
L.UNT   = ******  GX          10-236      10-244       12-354       12-362
MS.CIR  = ******  GX          17-660
MS.EXT  = ******  GX          17-653      19-738       19-749
MS.INC  = ******  GX          17-653      19-738       19-749
NXTLIN    001222 R            10-217      12-335      #14-483
OF$INS  = 000001             #5-69
OF$LOG  = 000100             #5-69
OF$OFF  = 000002             #5-69
OF$ON   = 000000             #5-69
OF$SMC  = 000200             #5-69
ONOFF     001566 R           #15-547
OP$ADD  = 000200             #5-69
OP$ALL  = 000200             #5-69
OP$CCS  = 000004             #5-69        10-211
OP$CIE  = 004000             #5-69
OP$CON  = 000400             #5-69
OP$COP  = 000001             #5-69
OP$COS  = 000100             #5-69        17-677
OP$DST  = 000200             #5-69
OP$DTE  = 000040             #5-69
OP$DUP  = 000002             #5-69
OP$EST  = 000100             #5-69
OP$EVE  = 010000             #5-69
OP$FIL  = 000200             #5-69
OP$GRP  = 000100             #5-69
OP$HOS  = 000001             #5-69
OP$HTM  = 000002             #5-69        17-672
OP$INA  = 004000             #5-69
OP$INC  = 001000             #5-69
```

```
                                                                  N 4
VNP        CREATED BY  MACRO  ON 29-JUN-85 AT 02:45     PAGE 4
```

```
**FILE**ID**VNTI







VV      VV   NN      NN   TTTTTTTTTT     IIIIII
VV      VV   NN      NN   TTTTTTTTTT     IIIIII
VV      VV   NN      NN       TT           II
VV      VV   NN      NN       TT           II
VV      VV   NNNN    NN       TT           II
VV      VV   NNNN    NN       TT           II
VV      VV   NN  NN  NN       TT           II
VV      VV   NN  NN  NN       TT           II
VV      VV   NN    NNNN       TT           II
VV      VV   NN    NNNN       TT           II
  VV  VV     NN      NN       TT           II           ....
  VV  VV     NN      NN       TT           II           ....
    VV       NN      NN       TT         IIIIII         ....
    VV       NN      NN       TT         IIIIII         ....





LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
001252  016701  000002G                  400$:   MOV     NTPSIZ+2,R1            ; GET RDB/SDB SIZE
001256                                           CALL    $ALL22                 ; TRY TO ALLOCATE
001262  103004                                   BCC     410$
001264                                           NTIERR  10                     ; NO ROOM IN NTPOOL

001274  010004                           410$:   MOV     R0,R4                  ; COPY BIAS
001276  012705  140000                           MOV     #.BASEB,R5             ; SET MAPPING DISPLACEMENT

001302                                   418$:   CALL    $SDBS                  ; ALLOCATE SMALL DATA BUFFERS
001306  103004                                   BCC     420$                   ; NO ERRORS
001310                                           NTIERR  12                     ; RESOURCE ERROR

001320                                   420$:   CALL    $RDBS                  ; ALLOCATE READ DATA BUFFERS
001324  103004                                   BCC     430$                   ; NO ERRORS
001326                                           NTIERR  11                     ; RESOURCE ERROR

001336                                   430$:
                                         ;
                                         ; LOAD PROCESSES INTO CORE
                                         ;
001336                                           CALL    $MFL                   ; LOAD WITH CALLS TO NTLOADER

                                         ;
                                         ; CHECK FOR DLX ONLY SYSTEM - IF TRUE, SET VF.DLX AND CALL $VACP TO START
                                         ;  AUX TIMERS
                                         ;
001342  012700  114224                           MOV     #^RXPT,R0              ; IS THERE AN XPT PROCESS?
001346  012701  000001                           MOV     #1,R1                  ; MUST CALL PARALLEL OVERLAY SEGMENT,
001352  000167  000000G                          JMP     $SWOVL                 ; ... JUMP TO ADDR IN ROOT TO SWITCH.
                                                                                ; ... SWOVL WILL CALL $FPDV
001356  006001                           $RET1:: ROR     R1                     ; GET BACK THE CARRY
001360  103007                                   BCC     $RET2                  ; FOUND ONE SO RETURN
001362  052767  000000G 000000G                  BIS     #VF.DLX,$VFLAG         ; CANT FIND ONE, MUST BE DLX ONLY
001370  012701  000002                           MOV     #2,R1                  ; MUST SWITCH OVERLAY SEGMENTS TO CALL
001374  000167  000000G                          JMP     $SWOVL                 ; ROUTINE TO START AUX TIMER SERVICE
                                                                                ; ... SWOVL WILL CALL $VACP
001400  042767  000000G 000000G  $RET2:: BIC     #VF.DLX,$VFLAG         ; NO LONGER MAKE EXCEPTION FOR DLX SYSTM
001406                                           RETURN
```

```
003534  016746  000014G                 MOV     $BFR+P.REL,-(SP) ; GET THE BASE ADDRESS OF THE PARTITION
003540  066716  000016G                 ADD     $BFR+P.SIZE,(SP) ; GET THE END ADDRESS OF THE PARTITION
003544  032767  000000G 000000G         BIT     #VF.SS,$VFLAG    ; 11S SYSTEM ?
003552  001416                          BEQ     19$              ; IF NO - BRANCH

003554  012705  000000G                 MOV     #SYSFDB,R5      ; GET FDB ADDRESS
003560  062705  000002G                 ADD     #F.HIBK+2,R5    ; POINT TO MAX VBN
003564  011505                          MOV     (R5),R5         ; GET MAX VBN
003566  162705  000002                  SUB     #2,R5           ; CALCULATE TOP OF MEMORY BIAS
003572  000241                          CLC                     ; ...
003574  006305                          ASL     R5              ; ...
003576  006305                          ASL     R5              ; ...
003600  006305                          ASL     R5              ; ...
003602  020526                          CMP     R5,(SP)+        ; IS ADDRESS WITHIN SYSTEM IMAGE FILE ?
003604  103006                          BHIS    20$             ; IF YES - BRANCH
003606  000403                          BR      22$

003610  022726  007600          19$:    CMP     #7600,(SP)+     ; IS THE PARTITION WITHIN THE 124K LIMIT ?
003614  103002                          BHIS    20$             ; IF HIS, YES .. OKAY, DO IT
003616  005067  000000G         22$:    CLR     TOPFLG          ; ELSE, DO BOTTOM-UP LOADING

003622  010105                  20$:    MOV     R1,R5           ; COPY PARTITION ADDRESS
003624                                  CALL    CRSUB           ; ATTEMPT TO ALLOCATE SUB-PARTITION
003630                          30$:    RETURN
```

ALLOCATE SUB-PARTITION OUT OF MAIN PARTITION

```
   1569 005010                                 PUTRC   R0,,PLGTH           ; WRITE INITIALIZED PCB TO SYSTEM IMAGE
   1570 005026  010501                         MOV     R5,R1               ; COPY MAIN PCB ADDRESS
   1571 005030  005002                         CLR     R2                  ; NO PREVIOUS SUB-PCB
   1572 005032  005046                         CLR     -(SP)               ; DITTO
   1573 005034                         10$:    GETRV   R1                  ; READ IN PCB
   1574 005044  066702  000014G                ADD     P.REL+$BFR,R2       ; GIVING BIAS OF NEXT HOLE IN PARTITION
   1575 005050  016703  000010G                MOV     P.SUB+$BFR,R3       ; GET NEXT SUB-PCB
   1576 005054  001450                         BEQ     20$                 ; IF ZERO, END OF LIST
   1577 005056                                 GETRV   R3                  ; READ IN PCB
   1578 005066  016704  000014G                MOV     P.REL+$BFR,R4       ; GET BIAS OF SUB-PARTITION
   1579 005072  160204                         SUB     R2,R4               ; GIVING SIZE OF HOLE
   1580 005074  166704  000000G                SUB     NTPSIZ,R4           ; IS THE HOLE BIG ENOUGH?
   1581 005100  103426                         BLO     15$                 ; IF MI, NO
   1582 005102  010116                         MOV     R1,(SP)             ; SAVE PREVIOUS SUB-PCB ADDRESS
   1583 005104                                 GETRV   R0                  ; READ IN PCB
   1584 005114  010267  000014G                MOV     R2,P.REL+$BFR       ; SET POSSIBLE BIAS
   1585 005120  010267  000000G                MOV     R2,PSTART           ; SAVE IT
   1586 005124  010367  000010G                MOV     R3,P.SUB+$BFR       ; SET POINTER TO NEXT SUB-PCB
   1587 005130                                 PUTRC                       ; WRITE UPDATED PCB
   1588 005134  005767  000000G                TST     TOPFLG              ; TOP-DOWN ALLOCATION?
   1589 005140  001453                         BEQ     40$                 ; IF EQ, NO
   1590 005142  060467  000014G                ADD     R4,P.REL+$BFR       ; ALLOCATE FROM TOP OF HOLE
   1591 005146  060467  000000G                ADD     R4,PSTART           ; SAVE BIAS OF SUB-PARTITION
   1592 005152                                 PUTRC                       ; WRITE UPDATED PCB
   1593 005156  010301                 15$:    MOV     R3,R1               ; COPY SUB-PCB ADDRESS
   1594 005160                                 GETRV   R1                  ; READ PCB
   1595 005170  016702  000016G                MOV     P.BLKS+$BFR,R2      ; GET SUB-PCB SIZE
   1596 005174  000717                         BR      10$                 ;
   1597 005176                         20$:    GETRV   R5                  ; READ IN MAIN PCB
   1598 005206  016704  000014G                MOV     P.REL+$BFR,R4       ; GET MAIN PARTITION BIAS
   1599 005212  066704  000016G                ADD     P.BLKS+$BFR,R4      ; GIVING BIAS OF END OF PARTITION
   1600 005216  160204                         SUB     R2,R4               ; GIVING SIZE OF LAST HOLE
   1601 005220  166704  000000G                SUB     NTPSIZ,R4           ; IS LAST HOLE BIG ENOUGH?
   1602 005224  103421                         BLO     40$                 ; IF MI, NO
   1603 005226  005767  000000G                TST     TOPFLG              ; TOP-DOWN ALLOCATION?
   1604 005232  001401                         BEQ     30$                 ; IF EQ, NO
   1605 005234  060402                         ADD     R4,R2               ; ALLOCATE FROM TOP OF HOLE
   1606 005236  010116                 30$:    MOV     R1,(SP)             ; SAVE PREVIOUS SUB-PCB ADDRESS
   1607 005240                                 GETRV   R0                  ; READ IN PCB
   1608 005250  010267  000014G                MOV     R2,P.REL+$BFR       ; SET SUB-PARTITION BIAS
   1609 005254  010267  000000G                MOV     R2,PSTART           ; SAVE IT
   1610 005260  010367  000010G                MOV     R3,P.SUB+$BFR       ; SET LINK TO NEXT SUB-PCB
   1611 005264                                 PUTRC                       ; REWRITE PCB
   1612 005270  012601                 40$:    MOV     (SP)+,R1            ; GET PREVIOUS SUB-PCB ADDRESS
   1613 005272  001426                         BEQ     101$                ; IF EQ, NO SPACE IN PARTITION
   1614 005274                                 GETRV   R1                  ; READ IN PCB
   1615 005304  010067  000010G                MOV     R0,P.SUB+$BFR       ; LINK PREVIOUS SUB-PCB TO OURS
   1616 005310                                 PUTRC                       ; WRITE PCB BACK TO SYSTEM IMAGE
   1617 005314  010001                         MOV     R0,R1               ; THIS IS NOW THE OFFICIAL PCB
   1618
   1619 005316  005767  000000G                TST     SBPOOL              ; SERVER BASE SYSTEM ?
   1620 005322  001411                         BEQ     45$                 ; IF NOT - BRANCH
   1621
   1622 005324  016767  000000G 000000G        MOV     NTPSAV,NTPSIZ       ; RESTORE POOL.. SIZE
   1623 005332  016700  000000G                MOV     PSTART,R0           ; GET STARTING ADDRESS OF SERVER BASE POOL
   1624 005336  012667  000000G                MOV     (SP)+,PSTART        ; RESTORE VARIABLE
   1625 005342  012667  000000G                MOV     (SP)+,PBLOX         ; RESTORE VARIABLE
```

```
                                        .SBTTL  STA$DF PARSE TABLES
                                ;+
                                ;       STATE TRANSITIONS FOR THE 'STA$DF' MACRO
                                ;-

 006324                                 STATE$  STADF
 006324                                  TRAN$   $NUMBR
 006324                                 STATE$
 006324                                  TRAN$   <',>
 006324                                 STATE$
 006324                                  TRAN$   $RAD50
 006324                                 STATE$
 006324                                  TRAN$   <',>,STADF1
 006324                                  TRAN$   $LAMDA,$EXIT,X.FLG1
 006324                                 STATE$  STADF1
 006324                                  TRAN$   '!,STADF1
 006324                                  TRAN$   %SF.ENA%,STADF1,,SF.ENA,MRKFL
 006324                                  TRAN$   %SF.MFL%,STADF1,,SF.MFL,MRKFL
 006324                                  TRAN$   %SF.SER%,STADF1,,SF.SER,MRKFL
 006324                                  TRAN$   $RAD50,STADF1
 006324                                  TRAN$   $LAMDA,$EXIT,X.FLG
```

```
  2397                                            .SBTTL  TPARS STATE TABLES - SER$DF
  2398                                    ;+
  2399                                    ; SER$DF STATE TABLES
  2400                                    ;-
  2401 006324                                     STATE$  SERDF
  2402 006324                                      TRAN$  $LAMDA,,SERINI
  2403
  2404 006324                                     STATE$
  2405 006324                                      TRAN$  $EOS,$EXIT,SEREND
  2406 006324                                      TRAN$  <',>,A
  2407 006324                                      TRAN$  $NUMBR,,NOAR
  2408
  2409 006324                                     STATE$
  2410 006324                                      TRAN$  <',>
  2411
  2412 006324                                     STATE$  A
  2413 006324                                      TRAN$  <',>,B
  2414 006324                                      TRAN$  $NUMBR,,NOAD
  2415
  2416 006324                                     STATE$
  2417 006324                                      TRAN$  $EOS,$EXIT,SEREND
  2418 006324                                      TRAN$  <',>
  2419
  2420 006324                                     STATE$  B
  2421 006324                                      TRAN$  <',>,C
  2422 006324                                      TRAN$  $NUMBR,,HSAR,SF.HST,SERFLG
  2423
  2424 006324                                     STATE$
  2425 006324                                      TRAN$  <',>
  2426
  2427 006324                                     STATE$  C
  2428 006324                                      TRAN$  <',>,D
  2429 006324                                      TRAN$  $NUMBR,,HSAD
  2430
  2431 006324                                     STATE$
  2432 006324                                      TRAN$  $EOS,$EXIT,SEREND
  2433 006324                                      TRAN$  <',>
  2434
  2435 006324                                     STATE$  D
  2436 006324                                      TRAN$  <',>,DEVIC
  2437 006324                                      TRAN$  $STRNG,,SERCR,SF.CIR,SERFLG
  2438
  2439 006324                                     STATE$
  2440 006324                                      TRAN$  <'->
  2441
  2442 006324                                     STATE$
  2443 006324                                      TRAN$  $NUMBR,,SERUNI
  2444
  2445 006324                                     STATE$
  2446 006324                                      TRAN$  !SERST,DEVIC
  2447 006324                                      TRAN$  $LAMDA,,SERSLN
  2448
  2449 006324                                     STATE$
  2450 006324                                      TRAN$  $EOS,$EXIT,SEREND
  2451 006324                                      TRAN$  <',>
  2452
  2453 006324                                     STATE$  DEVIC
```

```
                                          .SBTTL  SLT$DF ACTION ROUTINES
                                  ;
                                  ; CONTROLLER NUMBER (SLT$DF)
                                  ;
006604  116767  000000G 000060'  .SCTL:   MOVB    .PNUMB,CTL          ; SAVE NUMBER
006612                                    RETURN

                                  ;
                                  ; UNIT NUMBER (SLT$DF)
                                  ;
006614  016700  000000G          .SUNT:   MOV     $SLTMA,R0           ; GET SLT MAPPING TABLE ADDRESS
006620  016701  000000G                   MOV     $SLTNM,R1           ; GET NUMBER OF SLT'S
006624  001425                            BEQ     40$                 ; IF ZERO, LINE NOT FOUND !
006626  012002                   10$:     MOV     (R0)+,R2            ; GET NEXT SLT
006630  001421                            BEQ     30$                 ;; IF EMPTY, SKIP IT
006632  126267  000012  000060'           CMPB    L.CTL(R2),CTL       ; CONTROLLER NUMBER MATCH ?
006640  001015                            BNE     30$                 ; IF NE, NO
006642  126267  000013  000000G           CMPB    L.UNT(R2),.PNUMB    ; UNIT NUMBER MATCH ?
006650  001011                            BNE     30$                 ; IF NE, NO
006652  116202  000002                    MOVB    L.DDM(R2),R2        ; GET DDM PDV INDEX
006656  066702  000000G                   ADD     $PDVTA,R2           ; PLUS PDV TABLE ADDRESS
006662  011202                            MOV     (R2),R2             ; GET PDV ADDRESS
006664  026267  000004  000000G  20$:     CMP     Z.NAM(R2),PDVNAM    ; DO NAMES MATCH ?
006672  001406                            BEQ     50$                 ; IF EQ, YES
006674                           30$:     SOB     R1,10$              ; LOOP <SLT COUNT> TIMES
006700                           40$:     NTIERR  26                  ; ERROR - LINE NOT IN SYSTEM
006710  014002                   50$:     MOV     -(R0),R2            ; PICK UP SLT ADDRESS
006712  010267  000000G                   MOV     R2,$SLTA            ; SAVE IT
006716  056762  000040' 000000            BIS     MRKFL,L.FLG(R2)     ; UPDATE MARK-FOR-LOAD STATUS

006724  032762  000400  000000            BIT     #LF.BRO,L.FLG(R2)   ; ETHERNET LINE ?
006732  001413                            BEQ     55$                 ; IF NOT - BRANCH
006734  032762  002000  000000            BIT     #LF.ENA,L.FLG(R2)   ; MARK FOR ENABLE ?
006742  001416                            BEQ     60$                 ; IF NOT -BRANCH
006744  062702  000022                    ADD     #L.MPF,R2           ; POINT TO STATION FLAG
006750  052712  000100                    BIS     #SF.ENA,(R2)        ; SET FIRST STATION ENABLED
006754  016702  000000G                   MOV     $SLTA,R2            ; RESTORE SLT ADDRESS
006760  000407                            BR      60$                 ; CONTINUE

006762  032762  004000  000000   55$:     BIT     #LF.MFL,L.FLG(R2)   ; IS LINE MARKED-FOR-LOAD ?
006770  001003                            BNE     60$                 ; IF NE, YES .. OKAY
006772  042762  002000  000000            BIC     #LF.ENA,L.FLG(R2)   ; ELSE, MAKE SURE ENABLE ISN'T SET
007000                           60$:     RETURN
```

```
   3205
   3206 011614  022767  000077  000000G .RARE:  CMP     #63.,.PNUMB     ; CHECK LIMIT
   3207 011622  002421                          BLT     101$            ; IF ERROR - BRANCH
   3208 011624  016700  000000G                 MOV     .PNUMB,R0       ; COPY AREA
   3209
   3210         000012                          .REPT   10.
   3211                                         ASL     R0              ; POSITION AREA BITS
   3212                                         .ENDR
   3213 011654  000241                          CLC
   3214 011656  010067  000100'                 MOV     R0,REMARE       ; STORE AREA
   3215 011662  001401                          BEQ     101$            ; IF ERROR - BRANCH
   3216 011664                                  RETURN
   3217
   3218 011666  062716  000002          101$:   ADD     #2,(SP)         ; REJECT
   3219 011672                                  RETURN
   3220
   3221
   3222
```

```
                                        .SBTTL  ACTION ROUTINE: OBJECT NUMBER SEEN
                                ;+
                                ; *** - .ONUM - OBJECT NUMBER SEEN
                                ;
                                ; INPUTS:
                                ;       .PNUMB = THE OBJECT NUMBER
                                ;
                                ; OUTPUTS:
                                ;       TRANSITION ACCEPTED/REJECTED
                                ;
                                ;-

013242  116767  000000G 000063' .ONUM:: MOVB    .PNUMB,TYPE             ; STORE OBJECT NUMBER
013250  005067  000032'                 CLR     ONAME                   ; CLEAR THE OBJECT NAME
013254  005067  000034'                 CLR     ONAME+2                 ;          ...
013260  105067  000064'                 CLRB    MXCP                    ; CLEAR THE MAXIMUM NUMBER OF COPIES
013264  005767  000000G                 TST     .PNUMH                  ; IS OBJECT NUMBER A WORD VALUE ?
013270  001003                          BNE     10$                     ; IF NE, NO
013272  105767  000001G                 TSTB    .PNUMB+1                ; IS IT A BYTE VALUE ?
013276  001402                          BEQ     20$                     ; IF EQ, YES
013300  062716  000002          10$:    ADD     #2,(SP)                 ; ELSE, REJECT THE TRANSITION
013304                          20$:    RETURN
```

```
  4018
  4019                                        ;+
  4020                                        ;
  4021                                        ; SERALL - ALLOCATE SERVICE BLOCK
  4022                                        ;
  4023                                        ;-
  4024
  4025 014640  012701  000052                 SERALL: MOV     #S.LENG,R1          ; GET SIZE OF BLOCK NEEDED
  4026 014644                                         CALL    $ALPOL              ; ALLOCATE BLOCK FROM POOL..
  4027 014650  103004                                 BCC     10$                 ; IF SUCCESS - BRANCH
  4028
  4029 014652                                         NTIERR  33                  ; SERVICE BLOCK ALLOCATION FAILURE
  4030
  4031 014662  010067  000256'                10$:    MOV     R0,SERCUR           ; SAVE CURRENT SERVICE BLOCK ADDRESS
  4032
  4033 014666                                         CEACC$  R0                  ; MAP TO SERVICE BLOCK
  4034
  4035 014676  012702  000000G                        MOV     #$BFR,R2            ; GET BUFFER ADDRESS
  4036 014702  012701  000052                         MOV     #S.LENG,R1          ; SET BLOCK SIZE
  4037 014706  105022                         15$:    CLRB    (R2)+               ; ZERO OUT BLOCK
  4038 014710  005301                                 DEC     R1                  ; COUNT ONE LESS BYTE
  4039 014712  001375                                 BNE     15$                 ; IF MORE - BRANCH
  4040
  4041 014714  056767  000110' 000002G                BIS     SERFLG,S.FLAG+$BFR ; CLEAR FLAG WORD
  4042 014722  016767  000112' 000004G                MOV     SERNOD,S.ADD+$BFR ; GET NODE ADDRESS
  4043 014730  016767  000114' 000006G                MOV     SERHST,S.HST+$BFR ; GET HOST ADDRESS
  4044
  4045 014736  016767  000140' 000010G                MOV     SERADD,S.HAD+$BFR ; GET HARDWARE ADDRESS
  4046 014744  016767  000142' 000012G                MOV     SERADD+2,S.HAD+$BFR+2 ; ...
  4047 014752  016767  000144' 000014G                MOV     SERADD+4,S.HAD+$BFR+4 ; ...
  4048 014760  016767  000262' 000016G                MOV     SERSLT,S.CIR+$BFR ; GET SLN/STA PAIR
  4049
  4050 014766  016767  000120' 000020G                MOV     SERDEV,S.DEV+$BFR ; GET DEVICE NAME RAD50
  4051 014774  016767  000122' 000022G                MOV     SERPAS,S.PSS+$BFR ; GET SERVICE PASSWORD
  4052 015002  016767  000124' 000024G                MOV     SERPAS+2,S.PSS+$BFR+2 ; ...
  4053
  4054 015010  016767  000126' 000026G                MOV     SERPAS+4,S.PSS+$BFR+4 ; ...
  4055 015016  016767  000130' 000030G                MOV     SERPAS+6,S.PSS+$BFR+6 ; ...
  4056 015024  016767  000132' 000032G                MOV     SERDMP,S.DPA+$BFR ; GET DUMP ADDRESS
  4057 015032  016767  000134' 000034G                MOV     SERDMP+2,S.DPA+$BFR+2 ; ...
  4058 015040  016767  000136' 000036G                MOV     SERDCT,S.DPC+$BFR ; GET DUMP COUNT
  4059
  4060 015046                                         PUTAD   R0,#S.LENG          ; WRITE SERVICE BLOCK TO IMAGE
  4061
  4062 015064                                         SWTCHI  #$HMBUF             ; SWITCH INPUT BUFFER
  4063 015072                                         GETRV   $DECPT,#D$END       ; READ DECNET HOME BLOCK
  4064 015112                                         SWTCHI                      ; SWITCH TO DEFAULT INPUT BUFFER
  4065
  4066 015120  012702  000032G                        MOV     #$HMBUF+D$SER,R2 ; COPY SERVICE BLOCK LISTHEAD
  4067 015124  016700  000256'                        MOV     SERCUR,R0           ; COPY UNMAPPED ADDRESS OF BLOCK
  4068 015130  005712                                 TST     (R2)                ; LIST EMPTY ?
  4069 015132  001002                                 BNE     20$                 ; IF NO - BRANCH
  4070 015134  010201                                 MOV     R2,R1               ; IF YES - SET LISTHEAD AS PREVIOUS ENTRY
  4071 015136  000401                                 BR      25$                 ; CONTINUE
  4072 015140  011201                         20$:    MOV     (R2),R1             ; SET FIRST ENTRY AS PREVIOUS BLOCK
  4073 015142                                 25$:    CALL    QINS                ; INSERT INTO QUEUE
  4074
```

```

                                         .SBTTL  .FLBK0 - FILE NAME BLOCK PROCESSING

                                 ;+
                                 ;
                                 ; .FLBK0 - FILE NAME BLOCK PROCESSING ROUTINE
                                 ;
                                 ; INPUTS -
                                 ;                PARSED FILESPEC LOCATED IN DEVBUF,UICBUF,FILBUF
                                 ;
                                 ; OUTPUTS -
                                 ;                IF FNB CURRENTLY EXISTS - USAGE COUNT INCREMENTED
                                 ;                IF NOT - NEW FND CREATED AND LINKED
                                 ;
                                 ;-

                                         .ENABL  LSB

016372  012767  000040  000260' .FLBK0::MOV     #S.SEC,SEROFF   ; SAVE FILENAME LIST OFFSET
016400  052767  000400  000110'         BIS     #SF.SEC,SERFLG  ; SET SECONDARY BOOT FILE FLAG
016406  000433                          BR      5$              ; CONTINUE

016410  012767  000042  000260' .FLBK1::MOV     #S.TER,SEROFF   ; SAVE FILENAME LIST OFFSET
016416  052767  001000  000110'         BIS     #SF.TER,SERFLG  ; SET TERTIARY BOOT FILE FLAG
016424  000424                          BR      5$              ; CONTINUE

016426  012767  000044  000260' .FLBK2::MOV     #S.LOA,SEROFF   ; SAVE FILE NAME LIST OFFSET
016434  052767  002000  000110'         BIS     #SF.LOA,SERFLG  ; SET LOAD FILE FLAG
016442  000415                          BR      5$              ; CONTINUE

016444  012767  000046  000260' .FLBK3::MOV     #S.DIA,SEROFF   ; SAVE FILE NAME LIST OFFSET
016452  052767  004000  000110'         BIS     #SF.DIA,SERFLG  ; SET DIAGNOSTIC FILE FLAG
016460  000406                          BR      5$              ; CONTINUE

016462  012767  000050  000260' .FLBK4::MOV     #S.DUM,SEROFF   ; SAVE FILE NAME LIST OFFSET
016470  052767  010000  000110'         BIS     #SF.DUM,SERFLG  ; SET UPLINE DUMP FILE FLAG

016476                          5$:     CALL    SETBLK          ; SET UP LOCAL COPY OF FNB
016502                                  CALL    SRHFNB          ; SEARCH FOR EXISTING FNB
016506  103007                          BCC     10$             ; IF FOUND - BRANCH

016510                                  CALL    CRFNBK          ; IF NOT - CREATE NEW FNB
016514  103004                          BCC     10$             ; IF ALLOCATION SUCCESSFUL - BRANCH

016516                                  NTIERR  32              ; FILENAME BLOCK ALLOCATION FAILURE

016526                          10$:    SAVRG   <R3>            ; SAVE REG
016530  010003                          MOV     R0,R3           ; SAVE FILENAME BLOCK ADDRESS
016532                                  CEACC$  R0              ; MAP TO BLOCK
016542                                  GETAD   R0,#F.USE+2     ; READ FILENAME BLOCK
016560  105267  000002G                 INCB    $BFR+F.USE      ; BUMP USAGE COUNT
016564                                  PUTAD   R0,#F.USE+2     ; WRITE FNB TO IMAGE

016602  016700  000256'                 MOV     SERCUR,R0       ; COPY CURRENT SERVICE BLOCK ADDRESS
016606                                  CEACC$  R0              ; MAP TO BLOCK
016616                                  GETAD   R0,#S.LENG      ; READ SERVICE BLOCK
016634  056767  000110' 000002G         BIS     SERFLG,$BFR+S.FLAG ; SET FLAG WORD
```

```
VNTI - VNP NETWORK INITIALIZATI MACRO V05.03b Saturday 29-Jun-85 02:46  Page 110-3
Symbol table

KINAR6= 172354         LF.UNL= 020000         LO.TPA  000020         LT.DIR= 000010         L.OPDL  000102
KINAR7= 172356         LF.X2P= 000000         LO.TRB  000034         LT.LCL= 000001         L.OWNR  000021
KISAR0= 172340         LINSIZ= ****** GX      LO.TRI  000030         LT.LPL= 000002         L.REM   000006
KISAR5= 172352         LN.CLO= 000000         LO.UNT  000033         LT.NOT= 000040         L.RFC   000050
KISAR6= 172354         LN.DUM= 000005         LO.VCT  000014         LT.RSU= 000200         L.RLA   000004
KISAR7= 172356         LN.LOA= 000004         LO.VER  000042         LT.SLI= 000004         L.RNO   000022
KISDR0= 172300         LN.LOO= 000003         LR.ACC  000020         LT.TDA= 000100         L.RTQ   000060
KISDR6= 172314         LN.OAU= 000003         LR.ACO  000052         LX.ALI= 000017         L.RTYD  000055
KISDR7= 172316         LN.OFF= 000001         LR.ADD  000012         LX.CEX= 000004         L.RTYI  000057
KWORD   000032RG   004 LN.ON = 000000         LR.ALI  000002         LX.CIR= 000034         L.SEC   000064
K$$CNT= 177546         LN.OOP= 000004         LR.BLK  000010         LX.LIN= 000006         L.SEGZ  000076
K$$CSR= 177546         LN.OPE= 000001         LR.CTL  000004         LX.LOG= 000021         L.STA   000000
K$$LDC= 000000         LN.REF= 000002         LR.DES  000010         LX.NOD= 000015         L.TC    000042
K$$TPS= 000074         LN.SER= 000002         LR.HLP  000002         LX.OBJ= 000023         L.TIC   000043
LA.ACK= 100000         LN.STA= 000017         LR.LIN  000002         LX.PRO= 000005         L.TIPD  000013
LA.CRS= 020000         LN.SUB= 000360         LR.NAM  000012         L$$ASG= 000000         L.TIPI  000012
LA.MSK= 170000         LN.TRI= 000006         LR.PAS  000041         L$$DRV= 000000         L.TMRD  000054
LA.NAK= 110000         LOA     002020R    003 LR.PRO  000002         L$$P11= 000001         L.TMRI  000056
LA.NMS= 010000         L[illegible]T 001624R    003 LR.STS  000000         L$$11R= 000000         L.TYP   000001
LA.RES= 040000         LO.ADD  000022         LR.TRI  000006         L.COST  000015         L.UNT   000013
LA.WND= 004000         LO.ADR  000052         LR.TYP  000002         L.CSTA  000037         L.USA   000024
LBRA  = 000074         LO.CLS  000002         LR.UCB  000016         L.CTL   000012         L.USTA  000036
LDDLM   017636R        LO.CON  000006         LR.UIC  000020         L.CTR   000074         L.VER   000015
LDLINE  017516R        LO.COP  000010         LR.UNT  000005         L.CVA   177776         L.WIND  000040
LDPRO   020026R        LO.COS  000020         LST     011074R        L.DCR   000100         MACTYP= ****** GX
LD$LP = 000000         LO.CSR  000010         LS.ALI= 000016         L.DDM   000002         MAXCST= 001777
LD.ALI= 000030         LO.CTL  000032         LS.CEX= 000001         L.DDS   000004         MAXHOP= 000037
LD.CEX= 000024         LO.DES  000002         LS.CIR= 000033         L.DLC   000003         MA.CI = 000040
LD.CIR= 000035         LO.DTE  000047         LS.CXO= 010000         L.DLM   000006         MA.DA = 000000
LD.LIN= 000025         LO.EVT  000004         LS.CX5= 000400         L.DLS   000010         MA.IL = 000020
LD.LOG= 000031         LO.GRO  000024         LS.DLS= 100000         L.FLAG  000014         MC.ADD= 000200
LD.MOD= 000037         LO.HOS  000040         LS.ECH= 001000         L.FLG   000000         MC.CC = 000040
LD.NOD= 000027         LO.HTM  000032         LS.FCC= 000004         L.ILSQ  000052         MC.CI = 000020
LD.OBJ= 000032         LO.INA  000062         LS.FC0= 000001         L.ILTT  000066         MC.DC = 000100
LD.PRO= 000026         LO.INC  000060         LS.FC1= 000002         L.KRBA  000016         MC.DDM= 000004
LET   = 000100         LO.LDT  000046         LS.FDX= 004000         L.LDA   000032         MC.DI = 000060
LF.ACT= 100000         LO.LID  000030         LS.HDX= 010000         L.LEN = 000022         MC.DLC= 000003
LF.BRO= 000400         LO.LIN  000004         LS.HLP= 000012         L.LGST  013206R        MC.EVT= 001000
LF.BWT= 000007         LO.LTM  000034         LS.ILS= 100000         L.LIA   000034         MC.LLC= 000002
LF.DRD= 000004         LO.MAC  000022         LS.LIN= 000003         L.LLA   000002         MC.LOG= 000400
LF.ENA= 002000         LO.MDE  000024         LS.LMC= 000007         L.LNG   000124         MC.NO = 000000
LF.FRC= 000001         LO.NAM  000044         LS.LOG= 000020         L.LNO   000026         MC.NOD= 000010
LF.HFO= 000010         LO.NET= 000002         LS.MAK= 000020         L.LPT   000065         MC.OBJ= 000020
LF.HMF= 000040         LO.NNM  000022         LS.MNK= 000040         L.LSA   000030         MC.RC = 000140
LF.HSF= 000020         LO.OFL  000012         LS.MOD= 000036         L.LSFD  000046         MC.REM= 000040
LF.IRD= 000002         LO.ONM  000004         LS.NOD= 000014         L.LSFI  000044         MC.ROU= 000100
LF.LPB= 001000         LO.OUT  000056         LS.NTI= 000013         L.LTT   000062         MC.SLT= 000001
LF.MDC= 000100         LO.OWN  000026         LS.OBJ= 000022         L.MASQ  000070         MC.STA= 002000
LF.MFL= 004000         LO.PAR  000010         LS.OPT= 000400         L.MAST  000073         MD.BM = 000040
LF.MMF= 000200         LO.PRI  000012         LS.PRO= 000002         L.MASZ  000072         MD.EM = 000100
LF.MSF= 000100         LO.RET  000064         LS.RES= 000360         L.MPF   000022         MD.ILS= 000040
LF.MTP= 000020         LO.RPA  000030         LS.RSV= 000300         L.NIN   000020         MD.IM = 000020
LF.PAC= 000200         LO.SCR  000016         LS.TOP= 002000         L.NMST  000020         MF.ACK= 000004
LF.RDY= 040000         LO.SEG  000054         LS.TST= 000010         L.NSTA  000014         MF.CTL= 000010
LF.REA= 010000         LO.SKA  000014         LS.UNF= 020000         L.NXN   000016         MF.DAT= 000000
LF.SER= 000040         LO.SKN  000014         LS.XIT= 000011         L.NXTH  000010         MOVBUF  016016R
LF.TIM= 000010         LO.SNM  000035         LT.CCA= 000020         L.OPD   000103         MPAR  = 172100
```

```
VNCP      CREATED BY  MACRO  ON 29-JUN-85 AT 02:44     PAGE 5
SYMBOL CROSS REFERENCE                                 CREF    04.00

SYMBOL  VALUE              REFERENCES

LS.HDX  = 010000           #5-61
LS.HLP  = 000012           #5-61
LS.LIN  = 000003           #5-61
LS.LMC  = 000007           #5-61
LS.LOG  = 000020           #5-61       7-628        12-1428
LS.MOD  = 000036           #5-61       7-629
LS.NOD  = 000014           #5-61
LS.NTI  = 000013           #5-61
LS.OBJ  = 000022           #5-61
LS.OPT  = 000400           #5-61
LS.PRO  = 000002           #5-61
LS.TOP  = 002000           #5-61
LS.TST  = 000010           #5-61
LS.UNF  = 020000           #5-61
LS.XIT  = 000011           #5-61       6-248
LTRIBT  = 000041           #5-160      7-1208
LUIDTL  = 000035           #5-172      7-543
LUNUMT  = 000034           #5-153      7-1244
LX.ALI  = 000017           #5-61       7-627
LX.CEX  = 000004           #5-61
LX.CIR  = 000034           #5-61
LX.LIN  = 000006           #5-61
LX.LOG  = 000021           #5-61       7-628        12-1430
LX.NOD  = 000015           #5-61
LX.OBJ  = 000023           #5-61
LX.PRO  = 000005           #5-61
L.DDM   = ******  GX        11-1401
L.FLG   = ******  GX        7-1205     7-1275
MARKER    000104 R         #5-140      5-141        6-230
MAXID   = 000020           #5-78       7-273        7-277        13-1472      13-1484
MOVE      006522 R          7-282      13-1492     #14-1514
MSTRIB    001744 R         #5-176      5-176
MSUNUM    001677 R         #5-175      5-175
MXADDR  = 001777           #5-67       7-789        7-962        7-1014       7-1075
MXALIN  = 000006           #5-68       7-976
MXAREA  = 000077           #5-66       7-759
MXCN64    000667 R         #5-159      5-159        7-734
MXCOP   = 000077           #5-69       7-408
MXCST   = 000031           #5-70      #5-71         7-420
MXDSTN  = 000006           #5-72       7-264        7-984
MXDTEL  = 000020           #5-73       7-495
MXGRPN  = 000006           #5-74       7-560
MXLN64    000257 R         #5-150      5-150        7-746
MXNODN  = 000006           #5-77       7-1090       7-1097
MXPSCN  = 000006           #5-75       7-851
MXSNKN  = 000050           #5-76       7-1155
NAMETL    002201 R         #5-181      5-181        7-266        7-285        7-564        7-859        7-1116
NODID     000072 R         #5-128     *7-792       *7-793        7-994        7-1030       7-1035       7-1050       7-1089
NODNAM    004744 R          7-1041     7-1091      #7-1099
NSLINE    002016 R         #5-177      5-177        7-1209       7-1245       7-1280
NSTERM    001642 R         #5-174      5-174        7-913
OCOPTL    001305 R         #5-167      5-167        7-413
```

```
    147 000120  103774                      BCS     10$             ; USER CANNOT TYPE INTELLIGIBLY
    148 000122                              CALL    PROC            ; PROCESS REQUEST
    149 000126  000771                      BR      10$             ; IF NO - BRANCH
    150                                                             ; ELSE - EXIT
    151
```

```
                                          .SBTTL  $SPRO - SET PROCESS
                                  ;+
                                  ; $SPRO - "SET PROCESS"
                                  ;
                                  ; INPUTS:
                                  ;       $PDVA=PDV ADDRESS
                                  ;       $NAME=RAD50 PDV NAME
                                  ;       $$INC=INCREMENT VALUE (OPTIONAL)
                                  ;       $$PAR=PARTITION NAME (OPTIONAL)
                                  ;
                                  ; OUTPUTS:
                                  ;       C-BIT=SUCCESS/FAILURE
                                  ;-
001614                            $SPRO:
001614  016700  000000G                   MOV     $PDVA,R0        ; GET PDV ADDRESS
001620  005760  000000G                   TST     Z.PCB(R0)       ; IS PROCESS CORE-RESIDENT?
001624  001006                            BNE     10$             ; IF NE, YES
001626                                    CALL    $XFER           ; TRANSFER PROCESS FROM DISK TO CORE
001632  103471                            BCS     30$             ; IF CS, ERROR (NOTHING TO CLEAN UP)
001634  052767  000000G 000000G           BIS     #FL.PX1,$FLAGS  ; INDICATE I/O TRANSFER OCCURRED
001642  016701  000000G           10$:    MOV     $PDVA,R1        ; GET PDV ADDRESS
001646  032761  000000G 000000G           BIT     #ZF.PSE,Z.FLG(R1) ; IS THIS DLM?
001654  001004                            BNE     15$             ; BR IF YES
001656  032767  000000G 000000G           BIT     #FL.LLC,$FLAGS  ; ARE WE LOADING AN LLC?
001664  001437                            BEQ     25$             ; IF EQ, NO
001666                            15$:    CALL    $TMP            ; SETUP LLC LINE TABLE
001672  103452                            BCS     101$            ; IF CS, FAILURE
001674  005767  000000G                   TST     $$DEV           ; DATA STRUCTURES INCLUDED?
001700  001403                            BEQ     20$             ; IF EQ, NO
001702                                    CALL    $NLDEV          ; SETUP DEVICE DATA STRUCTURES
001706  103444                            BCS     101$            ; IF CS, FAILURE

001710  016701  000000G           20$:    MOV     $PDVA,R1        ; GET PDV ADDRESS
001714  026127  000000G 114224            CMP     Z.NAM(R1),#^RXPT ; LOADING XPT?
001722  001020                            BNE     25$             ; BR IF NO

001724                                    CALL    $ROL            ; ALLOCATE ROL MATRIX
001730  103433                            BCS     101$            ; IF ERROR - BRANCH
001732                                    CALL    $CACH           ; ALLOCATE CACHE
001736  103430                            BCS     101$            ; IF ERROR - BRANCH
001740                                    CALL    $CKT            ; ALLOCATE CIRCUIT COUNTERS
001744  103425                            BCS     101$            ; IF ERROR - BRANCH
001746                                    CALL    $ALADJ          ; ALLOCATE ADJACENCY DATA BASE
001752  103422                            BCS     101$            ; IF ERROR - BRANCH

001754                                    CALL    $ALMHC          ; ALLOCATE MINHOPS/MIN COST AND NODE COUNTERS
001760  103417                            BCS     101$            ; BR IF FAILURE
001762  000407                            BR      26$             ; CONTINUE

001764  026127  000000G 017704    25$:    CMP     Z.NAM(R1),#^RECL ; LOADING ECL ?
001772  001003                            BNE     26$             ; BR IF NO
001774                                    CALL    $ALECL          ; ALLOCATE NODE COUNTERS
002000  103407                            BCS     101$

002002                            26$:
002002  016701  000000G                   MOV     $PDVA,R1        ; GET PDV ADDRESS
002006  152761  000000G 000000G           BISB    #ZF.MFL,Z.FLG(R1) ; FORCE "CLEAR PROCESS" TO REMOVE FROM CORE
```

```
                                                                N  4
VNP       CREATED BY  MACRO  ON 29-JUN-85 AT 02:45      PAGE 4
SYMBOL CROSS REFERENCE                                  CREF    04.00

SYMBOL  VALUE            REFERENCES

OP$KDS  = 002000         #5-69
OP$KDT  = 000400         #5-69
OP$KEV  = 001000         #5-69
OP$KGR  = 001000         #5-69
OP$KNT  = 010000         #5-69
OP$KSK  = 002000         #5-69
OP$LCT  = 000200         #5-69
OP$LNE  = 000040         #5-69
OP$LOC  = 000001         #5-69      #5-69
OP$LST  = 000002         #5-69
OP$LTM  = 000001         #5-69       17-671
OP$LVL  = 000010         #5-69
OP$MAC  = 000010         #5-69       17-674
OP$MCO  = 000002         #5-69
OP$MDE  = 000400         #5-69       18-713
OP$MLI  = 000004         #5-69
OP$MON  = 000100         #5-69
OP$MST  = 000020         #5-69
OP$NAM  = 000004         #5-69      #5-69      #5-69
OP$NET  = 004000         #5-69
OP$NLI  = 000002         #5-69
OP$NOD  = 000020         #5-69
OP$OUT  = 002000         #5-69
OP$OWN  = 000020         #5-69       17-675
OP$PAR  = 000010         #5-69
OP$PRI  = 000010         #5-69
OP$RET  = 010000         #5-69
OP$RPA  = 000010         #5-69
OP$SCO  = 000001         #5-69
OP$SEG  = 000400         #5-69
OP$SER  = 000400         #5-69       17-678
OP$SIN  = 000010         #5-69
OP$STA  = 000004         #5-69       17-673
OP$TPA  = 000020         #5-69
OP$TRI  = 000040         #5-69       17-676
OP$UCS  = 000020         #5-69
OP$USE  = 000002         #5-69
OP$VEC  = 000040         #5-69       10-211
OP$VER  = 000040         #5-69
OP$XAC  = 000001         #5-69
OP$XPR  = 000002         #5-69
OP$X5S  = 000004         #5-69
OP$X9S  = 000010         #5-69
PRCIR     000652 R        9-189     #12-325
PRI       001612 R      *15-527      15-544    #15-553
PRLIN     000202 R        9-187       9-188    #10-205
PROC      000134 R        7-148      #9-173
PRTBL     000164 R        9-175      #9-187
P.REL   = ******   GX    20-791
QF$ACT  = 000001         #5-69
QF$ALL  = 000002         #5-69
QF$KNO  = 000004         #5-69       10-206     10-219     10-253     12-326     12-337     12-382
```

Table of contents

```
     5-   53    MACRO CALLS
     6-  108    LOCAL DATA
     8-  206    NETWORK INITIALIZATION COLD START
     9-  530    NTIERR - NTINIT ERROR HANDLING
    10-  566    NTWARN - NTINIT WARNING HANDLING
    11-  670    UMR ALLOCATION
    12-  754    ALLOCATE CCBS ( AND EXTRAS FOR RDBS )
    13-  804    ALLOCATE RDBS
    14-  848    INITIALIZE SMALL DATA BUFFER POOL
    15-  892    RETURN DOUBLE-WORD ADDRESS OF NEXT FREE BUFFER
    16-  926    DEALLOCATE RESOURCES
    17- 1024    DYNAMICALLY ALLOCATE SUB-PARTITION
    18- 1106    STATE TRANSITIONS FOR FINDING MAXIMUM NODE ADDRESS
    19- 1137    STATE TRANSITIONS FOR ESTIMATING NTPOOL SIZE
    20- 1176    STATE TRANSITIONS FOR FINDING NETWORK POOL PARTITION
    24- 1325    ACTION ROUTINES FOR ESTIMATING NTPOOL SIZE
    26- 1512    ALLOCATE SUB-PARTITION OUT OF MAIN PARTITION
    27- 1641    CFGMFL - CETAB SCANNER FOR NETWORK INITIALISER
    27- 1642    DEFINE LOCAL DATA AREAS
    28- 1670    PERFORM CONFIGURATION FILE SCAN
    29- 1715    OPEN CONFIGURATION FILE
    30- 1733    SCAN CONFIG FILF FOR NTPOOL DEFINITIONS
    31- 1850    PROCESS TEXT STRING FROM FILE
    32- 1878    STATE TRANSITION TABLES FOR PARSING
    33- 1911    BUF$DF PARSE TABLES
    34- 1924    LLC$DF,DLC$DF,DDM$DF PARSE TABLES
    35- 1951    SLT$DF PARSE TABLES
    36- 2004    STA$DF PARSE TABLES
    37- 2026    ADD$DF PARSE TABLES
    38- 2033    NOD$DF PARSE TABLES
    39- 2065    OBJ$DF PARSE TABLES
    41- 2179    REM$DF PARSE TABLES
    42- 2205    EVT$DF PARSE TABLES
    43- 2239    LOG$ST PARSE TABLES
    44- 2246    ROU$DF PARSE TABLES
    45- 2289    SUB-EXPRESSION PARSE TABLES
    48- 2397    TPARS STATE TABLES - SER$DF
    50- 2528    TPARS ACTION ROUTINES
    50- 2530    ACTION ROUTINE: BUF$DF MACRO SEEN
    51- 2545    ACTION ROUTINE: BUF$DF ARGUMENT MISSING
    52- 2560    ACTION ROUTINE: BUF$DF ARGUMENT SEEN
    53- 2575    BUF$DF ARGUMENT CONTROL TABLE
    54- 2595    ACTION ROUTINE: END OF BUF$DF SOURCE LINE
    55- 2614    ACTION ROUTINE: PROCESS NAME FOUND
    56- 2628    ACTION ROUTINE: 'MARKED FOR LOAD' FLAG SEEN
    57- 2650    ACTION ROUTINE: SLT$DF MACRO SEEN
    58- 2663    ACTION ROUTINE: STA$DF MACRO SEEN
    59- 2673    SLT$DF ACTION ROUTINES
    60- 2717    COUNTER TIMER ACTION ROUTINE
    61- 2744    ACTION ROUTINE: NODE NAME SEEN
    62- 2782    ACTION ROUTINE: NODE NUMBER SEEN
    63- 2829    ACTION ROUTINE: HOST NODE ADDRESS SEEN
    64- 2876    ACTION ROUTINE: SYSTEM ID SEEN
    65- 2901    ACTION ROUTINE: RIGHT ANGLE BRACKET SEEN
    66- 2974    ACTION ROUTINE: STATION FLAGS SEEN
    68- 3117    ACTION ROUTINE: REMOTE NODE NAME SEEN
    69- 3148    ACTION ROUTINE: REMOTE ADDRESS FOUND
```

```
                                                 .SBTTL  NTIERR - NTINIT ERROR HANDLING
                                         ;+
                                         ; *** - NTIERR - ERROR HANDLING
                                         ;
                                         ;       THIS ROUTINE FIELDS ALL ERRORS IN NTINIT. THE ERROR
                                         ;       CODE ON THE TOP TO THE STACK IS USED TO INDEX IN THE
                                         ;       TABLE 'ERRTAB' TO THE ADDRESS AND LENGTH OF THE ERROR
                                         ;       STRING. ALL RESOURCES ARE DEALOCATED.
                                         ;-


001410                                   NTIERR: CALL    $DALCT                 ; DEALLOCATE RESOURCES
001414  012767  001734' 000400                   MOV     #PREFIX,QIO+Q.IOPL ; ADDRESS OF PREFIX MESSAGE
001422  012767  000016  000374                   MOV     #PRELEN,QIO+Q.IOPL+2 ; LENGTH OF PREFIX MESSAGE
001430  012767  000044  000370                   MOV     #'$,QIO+Q.IOPL+4 ; LINE FEED
001436                                           DIR$    #QIO                   ; OUTPUT PREFIX MESSAGE
001444  011600                                   MOV     (SP),R0                ; GET INDEX VALUE ( ERROR CODE )
001446  042700  177400                           BIC     #^C<377>,R0            ; CLEAR FLAG BYTE
001452  006300                                   ASL     R0
001454  006300                                   ASL     R0                     ; CONVERT TO DOUBLE-WORD INDEX
001456  062700  002032'                          ADD     #ERRTAB-4,R0           ; POINT AT POINTER
001462  012067  000334                           MOV     (R0)+,QIO+Q.IOPL ; ADDRESS OF MESSAGE
001466  011067  000332                           MOV     (R0),QIO+Q.IOPL+2 ; LENGTH OF MESSAGE
001472  012767  000053  000326                   MOV     #'+,QIO+Q.IOPL+4 ; CARRIAGE RETURN
001500                                           DIR$    #QIO
001506  032716  000400                           BIT     #400,(SP)              ; DUMP LINE IMAGE ?
001512  001414                                   BEQ     100$
001514  016767  000000G 000300                   MOV     $CLBUF,QIO+Q.IOPL ; ADDRESS OF BUFFER
001522  016767  000000G 000274                   MOV     LINSIZ,QIO+Q.IOPL+2 ; BUF SIZE
001530  012767  000040  000270                   MOV     #40,QIO+Q.IOPL+4 ; LINE FEED, CARRIAGE RETURN
001536                                           DIR$    #QIO
001544  016706  000000G                  100$:   MOV     SPSAV,SP               ; RESTORE STACK POINTER
001550  000261                                   SEC                            ; SET C-BIT

001552                                           RETURN
```

```
   1106                                                  .SBTTL  STATE TRANSITIONS FOR FINDING MAXIMUM NODE ADDRESS
   1107
   1108 003632                                           ISTAT$  IBUF0,KBUF0
   1109
   1110 003632                                           STATE$
   1111 003632                                            TRAN$   %ROU$DF%,,,1,CFSFLG
   1112 003632                                           STATE$
   1113 003632                                            TRAN$   $NUMBR,,.R1
   1114 003632                                           STATE$
   1115 003632                                            TRAN$  <',>
   1116 003632                                           STATE$
   1117 003632                                            TRAN$  $NUMBR
   1118 003632                                           STATE$
   1119 003632                                            TRAN$  <',>
   1120 003632                                           STATE$
   1121 003632                                            TRAN$  $NUMBR
   1122 003632                                           STATE$
   1123 003632                                            TRAN$  <',>
   1124 003632                                           STATE$
   1125 003632                                            TRAN$  $NUMBR
   1126 003632                                           STATE$
   1127 003632                                            TRAN$  <',>
   1128 003632                                           STATE$
   1129 003632                                            TRAN$  $NUMBR,,.R2
   1130 003632                                           STATE$
   1131 003632                                            TRAN$  <',>
   1132 003632                                           STATE$
   1133 003632                                            TRAN$  $NUMBR,$EXIT,.R3
   1134 003632                                           STATE$
   1135
```

```

005346                                  45$:    RETURN

005350  016701  000000G                 101$:   MOV     .PLGTH,R1               ; DE-ALLOCATE SUB-PCB
005354                                          CALL    $DEA16                  ; ...
005360  005767  000000G                         TST     SBPOOL                  ; SERVER BASE SYSTEM ?
005364  001407                                  BEQ     102$                    ; IF NOT - BRANCH

005366  016767  000000G 000000G                 MOV     NTPSAV,NTPSIZ           ; RESTORE POOL.. SIZE
005374  012667  000000G                         MOV     (SP)+,PSTART            ; RESTORE VARIABLE
005400  012667  000000G                         MOV     (SP)+,PBLOX             ; ...

005404  005001                          102$:   CLR     R1                      ; SIGNAL FAILURE
005406                                          RETURN
```

```
  2026                                          .SBTTL  ADD$DF PARSE TABLES
  2027                                   ;+
  2028                                   ;      STATE TRANSITIONS FOR THE 'ADD$DF' MACRO
  2029                                   ;-
  2030 006324                                   STATE$  ADDDF
  2031 006324                                    TRAN$   $NUMBR,$EXIT
```

```
   2454 006324                                       TRAN$  <',>,E
   2455 006324                                       TRAN$  $STRNG,,SERDV,SF.DEV,SERFLG
   2456
   2457 006324                                     STATE$
   2458 006324                                       TRAN$  $EOS,$EXIT,SEREND
   2459 006324                                       TRAN$  <',>
```

```
                                                .SBTTL  COUNTER TIMER ACTION ROUTINE


                                        ;
                                        ; COUNTER TIMER VALUE
                                        ;
 007002                                 .SCTIM:
 007002  005767  000000G                        TST     .PNUMH          ; LEGAL COUNTER TIMER VALUE
 007006  001004                                 BNE     101$            ; BR IF NO
 007010  016767  000000G 000000G                MOV     .PNUMB,$CTIM    ; STORE COUNTER TIMER VALUE
 007016                                         RETURN
 007020                                 101$:   NTIERR  27              ; ILLEGAL COUNTER TIMER VALUE



                                        ;
                                        ; BROADCAST ROUTER ADJACENCIES
                                        ;
 007030  022767  000041  000000G        .SBRA:  CMP     #33.,.PNUMB     ; LEGAL VALUE ?
 007036  002404                                 BLT     10$             ; IF NO - BRANCH
 007040  066767  000000G 000102'                ADD     .PNUMB,NBRA     ; UPDATE TOTAL
 007046                                         RETURN

 007050  062716  000002                 10$:    ADD     #2,(SP)         ; ILLEGAL VALUE
 007054                                         RETURN


```

```
   3224                                                  .SBTTL  ACTION ROUTINE: MAXIMUM NUMBER OF NODES SEEN
   3225                                          ;+
   3226                                          ; *** - .RADDR - MAXIMUM NUMBER OF NODES SEEN
   3227                                          ;
   3228                                          ; INPUT:
   3229                                          ;       .PNUMB = MAXIMUM NUMBER OF NODES
   3230                                          ;
   3231                                          ;-
   3232
   3233 011674                                   .RADDR:
   3234 011674  022767  001777  000000G                  CMP     #1023.,.PNUMB           ; CHECK HIGH LIMIT
   3235 011702  002003                                   BGE     10$                     ; IF LEGAL VALUE - BRANCH
   3236 011704  062716  000002                           ADD     #2,(SP)                 ; IF NOT - REJECT
   3237 011710                                           RETURN
   3238
   3239 011712                                   10$:    GETRV   $DECPT,#D$END           ; READ DECNET HOME BLOCK
   3240
   3241 011732  016767  000000G 000060G                  MOV     .PNUMB,$BFR+D$NN        ; COPY MAX NODE VALUE
   3242
   3243 011740                                           PUTRC                           ; WRITE BACK DECNET HOME BLOCK
   3244
   3245 011744                                           RETURN
```

```
                                                .SBTTL  ACTION ROUTINE: OBJECT NAME SEEN
                                        ;+
                                        ; *** - .ONAME - OBJECT NAME SEEN
                                        ;
                                        ; INPUTS:
                                        ;       .PSTPT = POINTER TO THE OBJECT NAME STRING
                                        ;       .PSTCN = COUNT OF CHARACTERS IN THE OBJECT NAME STRING
                                        ;
                                        ; OUTPUTS:
                                        ;       TRANSITION IS ACCEPTED/REJECTED
                                        ;
                                        ;-

 013306  016700  000000G                .ONAME::MOV     .PSTPT,R0       ; GET ADDRESS OF ASCII TASK NAME
 013312  010701                                 MOV     PC,R1           ; PERIODS ARE ACCEPTABLE
 013314                                         CALL    $CAT5           ; CONVERT TASK NAME TO RAD50
 013320  010167  000032'                        MOV     R1,ONAME        ; STORE THE OBJECT NAME
 013324  103405                                 BCS     10$             ; IF CS, LESS THAN 3 CHARS CONVERTED
 013326  010701                                 MOV     PC,R1           ; CONVERT 2ND HALF OF NAME
 013330                                         CALL    $CAT5           ; ...
 013334  010167  000034'                        MOV     R1,ONAME+2      ; STORE 2ND HALF OF TASK NAME
 013340  022767  000006  000000G 10$:           CMP     #6,.PSTCN       ; IS THE TASK NAME TOO BIG ?
 013346  103002                                 BHIS    20$             ; IF HIS, NO
 013350  062716  000002                         ADD     #2,(SP)         ; REJECT TRANSITION
 013354                                 20$:    RETURN
```

```
   4075 015146                                        SWTCHO  #$HMBUF           ; SWITCH OUTPUT BUFFER
   4076 015154                                        PUTRC   $DECPT,#D$END     ; WRITE HOME BLOCK TO IMAGE
   4077 015174                                        SWTCHO                    ; RESET BUFFER TO DEFAULT
   4078
   4079 015202                                        RETURN
   4080
   4081
   4532 016642  012701  000000G                     MOV     #$BFR,R1         ; COPY BUFFER ADDRESS
   4533 016646  066701  000260'                     ADD     SEROFF,R1        ; GET ADDRESS OF FNB POINTER
   4534 016652  010311                              MOV     R3,(R1)          ; SET FNB ADDRESS
   4535
   4536 016654                                      PUTAD   R0,#S.LENG       ; WRITE UPDATED SERVICE BLOCK TO IMAGE
   4537 016672                                      RESRG   <R3>             ; RESTORE REG
   4538
   4539 016674  012767  000266' 000334' SETST:: MOV     #FNBUFF,STRPT    ; RESET START OF BUFFER
   4540 016702  005067  000204'                     CLR     STRSIZ           ; RESET STRING COUNT
   4541 016706  005067  000264'                     CLR     FNBLN            ; RESET FILE NAME LENGTH
   4542
   4543 016712  012700  000242'                     MOV     #DEVBUF,R0       ; INIT DEVICE BUFFER
   4544 016716  012701  000010                      MOV     #DVSZ,R1         ; SIZE OF BUFFER
   4545 016722                                      CALL    INIBUF           ; INIT BUFFER
   4546
   4547 016726  012700  000206'                     MOV     #UICBUF,R0       ; INIT UIC BUFFER
   4548 016732  012701  000014                      MOV     #UISZ,R1         ; SIZE OF BUFFER
   4549 016736                                      CALL    INIBUF           ; INIT BUFFER
   4550
   4551 016742  012700  000222'                     MOV     #FILBUF,R0       ; INIT FILE NAME BUFFER
   4552 016746  012701  000020                      MOV     #FLSZ,R1         ; SIZE OF BUFFER
   4553 016752                                      CALL    INIBUF           ; INIT BUFFER
   4554
   4555 016756                                      RETURN
   4556
   4557                                             .DSABL  LSB
   4558
   4559
   4560                                     ;+
   4561                                     ; GETSTR - STORE PARSED STRING
   4562                                     ;
   4563                                     ; INPUTS -
   4564                                     ;          .PSTPT - ADDRESS OF PARSED STRING
   4565                                     ;          .PSTCN - SIZE OF PARSED STRING
   4566                                     ;
   4567                                     ;-
   4568
   4569 016760  016700  000000G             GETSTR::MOV     .PSTPT,R0        ; GET ADDRESS OF STRING
   4570 016764  112077  000334'             10$:    MOVB    (R0)+,@STRPT     ; STORE BYTE
   4571 016770  005267  000334'                     INC     STRPT            ; INC POINTER
   4572 016774  005267  000204'                     INC     STRSIZ           ; COUNT ONE MORE CHARACTER
   4573 017000  005367  000000G                     DEC     .PSTCN           ; COUNT ONE LESS CHARACTER
   4574 017004  001367                              BNE     10$              ; IF MORE - BRANCH
   4575
   4576 017006                                      RETURN
   4577
   4578                                     ;+
   4579                                     ; DEV - STORE DEVICE NAME
   4580                                     ;
   4581                                     ; INPUTS -
   4582                                     ;          STRAD - CURRENT STRING ADDRESS
   4583                                     ;          STRSIZ - STRING SIZE
   4584                                     ;
   4585                                     ; OUTPUTS -
   4586                                     ;          DEVBUF CONTAINS DEVICE SRING
   4587                                     ;-
   4588
```

B	1	
C	1	
D	1	
E	1	
F	1	
G	1	
H	1	
I	1	
J	1	
K	1	
L	1	
M	1	
N	1	
B	2	
C	2	
D	2	
E	2	
F	2	
G	2	
H	2	VNP
I	2	
J	2	
K	2	
L	2	
M	2	
N	2	
B	3	
C	3	
D	3	
E	3	
F	3	
G	3	
H	3	
I	3	
J	3	
K	3	
L	3	
M	3	
N	3	
B	4	
C	4	
D	4	
E	4	
F	4	
G	4	
H	4	
I	4	
J	4	
K	4	
L	4	
M	4	
N	4	
B	5	
C	5	
D	5	
E	5	
F	5	VNSD
G	5	
H	5	
I	5	
J	5	
K	5	
L	5	
M	5	VNTI
N	5	
B	6	
C	6	
D	6	
E	6	
F	6	
G	6	
H	6	
I	6	
J	6	
K	6	
L	6	
M	6	
N	6	
B	7	
C	7	
D	7	
E	7	
F	7	
G	7	
H	7	
I	7	
J	7	
K	7	
L	7	
M	7	
N	7	
B	8	
C	8	
D	8	
E	8	
F	8	
G	8	
H	8	
I	8	
J	8	
K	8	
L	8	
M	8	
N	8	
B	9	
C	9	
D	9	
E	9	
F	9	
G	9	
H	9	
I	9	
J	9	
K	9	
L	9	
M	9	
N	9	
B	10	
C	10	
D	10	
E	10	
F	10	
G	10	
H	10	
I	10	
J	10	
K	10	
L	10	
M	10	
N	10	
B	11	
C	11	
D	11	
E	11	
F	11	
G	11	
H	11	
I	11	
J	11	
K	11	
L	11	
M	11	
N	11	
B	12	
C	12	
D	12	
E	12	
F	12	
G	12	
H	12	
I	12	
J	12	
K	12	
L	12	
M	12	
N	12	
B	13	
C	13	
D	13	
E	13	
F	13	
G	13	
H	13	
I	13	
J	13	
K	13	
L	13	
M	13	
N	13	
B	14	
C	14	
D	14	
E	14	
F	14	
G	14	
H	14	
I	14	
J	14	
K	14	
L	14	
M	14	
N	14	
B	15	
C	15	
D	15	
E	15	
F	15	
G	15	
H	15	
I	15	
J	15	
K	15	
L	15	
M	15	
N	15	
B	16	
C	16	
D	16	
E	16	
F	16	
G	16	
H	16	
I	16	
J	16	
K	16	
L	16	
M	16	